UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E INFORMÁTICA INDUSTRIAL

RICARDO DE SOUZA COSTA

DRM PARA PROTEÇÃO DE CONTEÚDO DE VOD UTILIZANDO O SGX: UMA ABORDAGEM DE AVALIAÇÃO DE DESEMPENHO DE DESCRIPTOGRAFIA VOLTADO PARA *SET-TOP BOXES*

DISSERTAÇÃO

CURITIBA

2019

RICARDO DE SOUZA COSTA

DRM PARA PROTEÇÃO DE CONTEÚDO DE VOD UTILIZANDO O SGX: UMA ABORDAGEM DE AVALIAÇÃO DE DESEMPENHO DE DESCRIPTOGRAFIA VOLTADO PARA *SET-TOP BOXES*

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de "Mestre em Ciências" – Área de Concentração: Telecomunicações e Redes.

Orientadora: Prof.ª Dr.ª Keiko Verônica Ono Fonseca

Co-orientador: Prof. Dr. Marcelo de Oliveira Rosa

CURITIBA

2019

Dados Internacionais de Catalogação na Publicação

Costa, Ricardo de Souza

DRM para proteção de conteúdo de VOD utilizando o SGX [recurso eletrônico]: uma abordagem de avaliação de desempenho de descriptografia voltado para *Set-Top Boxes* / Ricardo de Souza Costa. -- 2019.

1 arquivo eletrônico (74 f.): PDF; 1,62 MB.

Modo de acesso: World Wide Web.

Texto em português com resumo em inglês.

Dissertação (Mestrado) - Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial. Área de Concentração: Telecomunicações e Redes, Curitiba, 2019.

Bibliografia: f. 70-73.

1. Engenharia elétrica - Dissertações. 2. Intel Software Guard Extensions. 3. Video sob demanda. 4. Vídeo digital - Medidas de segurança. 5. Gerenciamento de direitos digitais. 6. Proteção de dados. 7. HTTP (Protocolo de rede de computadores). 8. Transmissão ao vivo. 9. Televisão interativa - Equipamentos e acessórios. 10. Criptografia de dados (Computação). 11. Desempenho - Avaliação. 12. Métodos de simulação. I. Fonseca, Keiko Verônica Ono, orient. II. Rosa, Marcelo de Oliveira, coorient. III. Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial. IV. Título.

CDD: Ed. 23 -- 621.3

Biblioteca Central do Câmpus Curitiba - UTFPR
Bibliotecária: Luiza Aquemi Matsumoto CRB-9/794

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Diretoria de Pesquisa e Pós-Graduação

TERMO DE APROVAÇÃO DE DISSERTAÇÃO Nº_____

A Dissertação de Mestrado intitulada **DRM para Proteção de Conteúdo de VOD utilizando o SGX: Uma abordagem de Avaliação de Desempenho de Descriptografia voltado para Set-Top Boxes**, defendida em sessão pública pelo candidato Ricardo de Souza Costa, no dia 12 de dezembro de 2019, foi julgada para a obtenção do título de Mestre em Ciências, área de concentração Telecomunicações e Redes, e aprovada em sua forma final, pelo Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial.

BANCA EXAMINADORA:

Prof(a). Dr(a). Keiko Verônica Ono Fonseca - Presidente - UTFPR

Prof. Dr. Daniel Fernando Pigatto - UTFPR

Prof. Dr. António Navarro - Universidade de Aveiro (Portugal)

A via original deste documento encontra-se arquivada na Secretaria do Programa, contendo a assinatura da Coordenação após a entrega da versão corrigida do trabalho.

Curitiba, _____de dezembro de 2019_.

Carimbo e Assinatura do(a) Coordenador(a) do Programa

AGRADECIMENTOS

A Deus, primeiro de tudo, pela vida e bênçãos recebidas.

A minha esposa Jocilene, por todo o amor, carinho e incentivo.

Aos meus filhos Luiz Henrique e Guilherme, pela compreensão e pela paciência por passar o meu tempo na dedicação na elaboração deste trabalho, ao invés de participar de suas brincadeiras.

A minha orientadora Prof.ª Dr.ª Keiko Verônica Ono Fonseca que acreditou em meu potencial em me conduzir com seus conhecimentos para o sucesso deste trabalho.

Ao meu co-orientador Prof. Dr. Marcelo de Oliveira Rosa, que me orientou desde o início desta jornada com a pesquisa e auxiliou na implementação do algoritmo utilizando o SDK do Intel SGX.

Ao Prof. Dr. Daniel Fernando Pigatto que me orientou com a análise de desempenho e obtenção dos resultados para a conclusão deste trabalho além de disponibilizar o seu tempo para apresentar meu artigo científico no SbSEG 2018 em Natal-RN.

Aos participantes da pesquisa e colegas pela ajuda.

A UTPFR e ao CPGEI por terem me proporcionado o ambiente necessário para aquisição do conhecimento científico e elaboração deste trabalho.

A MCTI/RNP pelo apoio financeiro ao projeto EU-BR SecureCloud (MCTI/RNP 3ª chamada coordenada) no Brasil.

“A insistência e a coragem forjam a persistência que dá luz ao caminho do êxito da perfeição.”

Julio Aukay

RESUMO

DE SOUZA COSTA, Ricardo. DRM PARA PROTEÇÃO DE CONTEÚDO DE VOD UTILIZANDO O SGX: UMA ABORDAGEM DE AVALIAÇÃO DE DESEMPENHO DE DESCRIPTOGRAFIA VOLTADO PARA *SET-TOP BOXES*. 74 f. Dissertação – Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2019.

O Vídeo sob Demanda ou simplesmente VOD é um serviço de *streaming* de mídia que exige uma alta largura de banda em redes de IPTV, mas por outro lado, é também uma enorme fonte de receita para provedores de conteúdo, emissoras e provedores de infraestrutura. Os clientes podem selecionar conteúdo de TV, como filmes e programas, a partir de uma ampla biblioteca de mídia digital, e não estão mais vinculados a um EPG (*Electronic Program Guide*). No entanto, o serviço de VOD deve disponibilizar o *streaming* de um conteúdo escolhido apenas para clientes autorizados, ou seja, o sistema deve proteger a transferência de dados e armazenamento, a fim de evitar violações de direitos autorais ou concorrência empresarial injusta com conteúdo adquirido ilegalmente. E conforme o problema em questão, é implementado uma solução de DRM (*Digital Rights Management*) baseado na tecnologia do Intel SGX (*Software Guard Extensions*), com foco no dispositivo do cliente, para avaliar o desempenho de descriptografia de vídeo, com resoluções em varreduras progressivas de 480, 720, 1080 e 2160 em uma *enclave*. O resultado mostra que o desempenho de descriptografia de um vídeo de resolução de 2160p, com um *buffer* de entrada de dados de 32kB na *enclave*, por exemplo, obtém-se a média de 10*ms*. E consequentente, esse resultado atende aos requisitos dos equipamentos de decodificadores de vídeo atuais do mercado.

Palavras-chave: SGX, VOD, DRM, Proteção de Conteúdo, HLS, STB

ABSTRACT

DE SOUZA COSTA, Ricardo. DRM FOR VOD CONTENT PROTECTION USING SGX: A DECRYPTION PERFORMANCE EVALUATION APPROACH FOR SET-TOP BOXES. 74 f. Dissertação – Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2019.

Video on Demand or simply VOD is a streaming media service that requires high bandwidth on IPTV networks, but it is also a huge source of revenue for content providers, broadcasters and infrastructure providers. Customers can select TV media contents from a large digital media library and are no longer dependent on an EPG (Electronic Program Guide). However, the VOD service must make streaming content chosen only for personalized customers, it means that the system must protect data transfer and storage in order to prevent copyright infringements or unfair business competition with illegally acquired content. With this issue in evidence, a DRM (Digital Rights Management) solution based on Intel SGX (Software Guard Extensions) technology is implemented, focusing on the client device, to evaluate video decryption performance in progressive scanning of 480, 720, 1080 and 2160 video resolutions into an enclave. The result shows the decryption performance of an 2160p video resolution, with an input data buffer of 32kB into the enclave, for example, gets an average of 10ms. Consequently, this result is compatible with the requirements of video decoder devices available on the market.

Keywords: SGX, VOD, DRM, Content Protection, HLS, STB

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE SIGLAS

API	*Application Programming Interface*
ABR	*Adaptive Bit Rate*
AES	*Advanced Encryption Standard*
ARM	*Advanced RISC Machine*
BIOS	*Basic Input/Output System*
CAS	*Conditional Access System*
CDN	*Content Delivery Network*
CPU	*Central Processor Unit*
DASH	*Dynamic Adaptive Streaming over HTTP*
DRM	*Digital Rights Management*
DTV	*Digital Television*
EPG	*Electronic Program Guide*
FCC	*Federal Communications Comission*
GCC	*GNU Compiler Collection*
GCM	*Galois Counter Mode*
HAS	*HTTP-based Adaptive Streaming*
HD	*High Definition*
HLS	*HTTP Live Streaming*
HTTP	*Hypertext Transfer Protocol*
IPTV	*Internet Protocol Television*
LD	*Low Definition*
LTS	*Long Term Support*
NIST	*National Institute of Standards and Technology*
NTSC	*National Television System Committee*
OTT	*Over the Top*
PAL	*Phase Alternating Line*
PC	*Personal Computer*
PCR	*Platform Configuration Register*
QoS	*Quality of Services*
RAM	*Random-Access Memory*

SECAM	*Séquentiel Couleur à Mémoire*
SD	*Standard Definition*
SDK	*Software Development Kit*
SGX	*Software Guard Extensions*
SPM	*Secure Process Memory*
STB	*Set-Top Box*
TEE	*Trusted Execution Environment*
TPM	*Trusted Platform Module*
TV	Televisão
UHD	*Ultra High Definition*
VOD	*Video On Demand*

LISTA DE SÍMBOLOS

f ps	*frame per seconds*
GB	gigabyte
GHz	gigahertz
kB	quilobyte
MB	megabyte
ms	milissegundos
us	microssegundos

SUMÁRIO

1 INTRODUÇÃO

Atualmente, as empresas de telecomunicações que prestam serviços de TV estão buscando novas soluções que associem distribuição de conteúdo de TV a outros serviços interativos, por exemplo: compras *on-line*, jogos e aluguel de filmes, como recurso de conveniência a seus clientes. Um dos serviços mais utilizados é a modalidade de aluguel de vídeo (*Video on Demand* ou VOD) em combinação com um gravador de vídeo, permitindo aos clientes selecionarem e visualizarem filmes disponíveis a qualquer momento.

O serviço VOD é executado interativamente utilizando comandos simples através do controle remoto com ações de: reproduzir, pausar, retroceder ou avançar a mídia em questão (PELTONIEMI, 1995).

Com a alta demanda de vídeo pela *Internet*, estima-se que o tráfego de vídeo atingirá globalmente 77% de todo o tráfego de dados da *Internet* até 2021 (CISCO, 2017). À medida que os aplicativos de VOD crescem, também aumentam os problemas de segurança relacionados a infrações de direitos digitais ou acesso não autorizado a conteúdos de vídeos protegidos (AKHYAR et al., 2015).

Desde 2016, a FCC (*Federal Communications Comission*) elaborou a norma FCC16-18 com o objetivo de adotar regras para guiar os consumidores a escolherem como eles desejam acessar as suas programações de vídeo multicanal (TV, computadores ou dispositivos móveis). E consequentemente, os provedores de conteúdos e fabricantes de decodificadores de vídeo são obrigados a reforçarem a segurança de seus dispositivos e serviços de conteúdo de mídia contra a pirataria e violação de direitos autorais (FCC, 2016).

Os decodificadores ou STBs (*Set-Top Boxes*) são utilizados pelas operadoras de TV por assinatura há vários anos e sua finalidade é decodificar sinais digitais para aparelhos de TV. Com o tempo, suas capacidades e a flexibilidade aumentaram e agora muitas empresas usam STBs digitais com recursos especiais.

E para proteger o conteúdo e os direitos autorais, a maioria dos STBs possuem um tecnologia conhecida como CA (*Conditional Access*) e o DRM (*Digital Rights Management*), a saber:

- O DRM é uma abordagem sistemática à proteção de direitos autorais de mídia digital. O

objetivo do DRM é impedir a redistribuição não autorizada de mídia digital e restringir a maneira como os consumidores podem copiar o conteúdo que compraram (RAMIREZ, 2008);

- O CA, como o próprio nome sugere, fornece o serviço de controle do acesso à televisão digital para pessoas que são ou não assinantes. O objetivo final é simples e claro - criptografar o sinal do canal e descriptografá-lo apenas para os espectadores que atendem às condições necessárias. O equipamento e o *software* necessários para o sistema são fornecidos pelo provedor CAS (*Conditional Access System*), para que as emissoras possam incorporá-lo em seu próprio equipamento (ASI, 2017).

1.1 MOTIVAÇÃO

O vídeo digital é um conjunto de informações binárias que representa o conteúdo de vídeo, compactado ou não, que pode ser facilmente copiado, transferido ou armazenado. Essas são as principais vantagens do vídeo digital sobre o vídeo analógico. No entanto, o vídeo digital é intrinsecamente desprotegido, ou seja, ele pode ser facilmente visualizado por qualquer pessoa e duplicado diversas vezes. Assim, quando existir direito de propriedade envolvido, haverá consequente necessidade de proteção e/ou controle de acesso sob algumas condições definidas (DIEHL, 2012).

Atualmente, existem diversas tecnologias disponíveis no mercado focadas na proteção de vídeo digital, mas a grande parte deles é baseada apenas em *software*. Por outro lado, há uma proteção para vídeo digital seja baseada em *hardware*, chamado de TPM (*Trusted Platform Module*).

O TPM é um dispositivo de segurança baseado em *hardware* que trata da integridade e proteção de dados. O TPM protege o processo de inicialização do sistema, garantindo que não haja violação antes de liberar o controle do sistema para o sistema operacional. Um dispositivo TPM fornece armazenamento seguro para dados, como chaves e senhas de segurança. Além disso, um dispositivo TPM possui funções de criptografia, *hash* e detecção de violações de *hardware* e *software* (INTEL, 2017b).

Nos dias atuais já existem processadores que incorporam uma solução de TPM, mas as aplicações e sistemas operacionais instalados em equipamentos com suporte ao TPM raramente utilizam tal recurso (DIEHL, 2012).

Desse modo, a presente dissertação propõe uma solução simples de proteção de vídeo digital distribuído em serviços de VOD utilizando TPM baseado na tecnologia Intel SGX

(*Software Guard Extensions*). A viabilidade técnica desta solução para descriptografar um vídeo digital por *hardware* é analisada através do desempenho de descriptografia e a partir destes resultados, são discutidas as vantagens e desvantagens da solução.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

O objetivo geral deste trabalho é de propor uma solução DRM para VOD baseado na tecnologia do Intel SGX a ser instalado em STBs ou decodificadores de vídeo similares. A análise da viabilidade técnica, por sua vez, dar-se-á através da avaliação de desempenho de descriptografia de vídeo em modo seguro.

1.2.2 OBJETIVOS ESPECÍFICOS

Para alcançar os objetivos específicos deste trabalho, os seguintes passos foram estabelecidos e executados:

- Implementar a solução de criptografia e descriptografia de VOD utilizando o sistema de ABR (*Adaptive Bit Rate*), chamado de HLS (*HTTP Live Streaming*) com o Intel SGX;
- Calcular o tempo gasto durante todo processo de descriptografia utilizando a biblioteca de criptografia do Intel SGX;
- Calcular o tempo gasto durante todo processo de descriptografia utilizando a biblioteca de criptografia do OPENSSL;
- Utilizar o algoritmo de criptografia AES-GCM (*Advanced Encryption Standard - Galois Counter Mode*) que já está disponível nativamente no SDK (*Software Development Kit*) do Intel SGX;
- Comparar a eficiência do algoritmo do AES-GCM tanto no SDK do Intel SGX quanto no OPENSSL na descriptografia apenas;
- Avaliar a viabilidade de integração da tecnologia SGX em aplicações de STBs, especificamente em serviços de VOD.

1.3 ESTADO DA ARTE

Nesta seção são citados alguns trabalhos relacionados pertinentes a este trabalho, que descrevem uma solução de DRM baseada em TPM ou associada a um serviço de VOD.

YANG et al. (2018) descrevem que os programadores podem escrever, compilar e instalar aplicações seguras baseadas no ARM *TrustZone* (ARM, 2016) disponível nos aparelhos de TV com o sistema operacional Tizen (SAMSUNG, 2018). Entretanto a utilização dos recursos de TPM dessa *SmartTV* requer uma API (*Application Programming Interface*) proprietária, que inclui um *pipeline* de mídia seguro dentre outros requisitos de proteção (MOVIELABS, 2015). Um estudo de caso referenciado nesse trabalho é uma aplicação de VOD utilizando essa solução. Eles criaram um DRM seguro chamado de *DRM Intrinsic Trusted Application*, que é executado dentro de um TPM.

O trabalho de Wu e Bao (2008) propõe uma implementação de DRM baseado em TPM adicionando um módulo chamado de SPM (*Secure Process Manager*) em *hardware* seguro como um componente de *kernel* para fins de gerenciamento de processos. O SPM tem o objetivo de fornecer uma interface confiável entre aplicação e o *kernel* seguro.

Yu et al. (2009) explica uma proposta de DRM utilizando TPM, chamado de TBDRM. As principais partes envolvidas no sistema são: o SERVIDOR DE CONTEÚDO, DISTRIBUIDOR DE LICENÇA e o CLIENTE. O SERVIDOR DE CONTEÚDO é responsável por gerar o conteúdo digital e gerar uma licença correspondente para o proprietário do vídeo. O DISTRIBUIDOR DE LICENÇA, por sua vez, gerencia a licença em nome do proprietário e distribui ao consumidor. O CLIENTE é a plataforma onde o consumidor assiste o conteúdo digital.

1.4 CONTRIBUIÇÕES

As principais contribuições desta dissertação são:

- Desenvolvimento de um sistema de medição de desempenho de descriptografia para o Intel SGX abordando a entrada e saída de dados em uma região de memória segura;
- Implementação de uma solução de criptografia e descriptografia para vídeos remultiplexados no padrão HLS, simulando um ambiente real de cliente de VOD utilizando a arquitetura de *software* do Intel SGX;

- Análise da viabilidade de integração de um TPM em um *player* cliente para o consumo de conteúdos de VOD, especificamente para *Set-Top Boxes*.

1.5 ORGANIZAÇÃO DA DISSERTAÇÃO

A dissertação está organizada da seguinte forma:

O capítulo 2 apresenta a fundamentação teórica referente às características de um vídeo digital, conceito do sistema de VOD, conceito de DRM, conceito de HLS, teoria de criptografia, criptografia de vídeo, as tecnologias mais usadas na segurança de STBs e de segurança de sistemas de VOD e o sistema de *hardware* seguro abrangendo a tecnologia do Intel SGX.

O capítulo 3 descreve a metodologia de avaliação de desempenho de descriptografia no Intel SGX e OPENSSL. Discute os diferentes tipos de vídeos utilizados na avaliação, a arquitetura do *software*, a arquitetura do sistema de medição de desempenho de descriptografia e a metodologia científica usada para medição e avaliação do desempenho da descriptografia.

O capítulo 4 apresenta os resultados e discussões das métricas propostas através de sua correlação com medidas na região insegura e região segura (*enclave*) com diversas resoluções de vídeos citados pela literatura.

O capítulo 5 descreve a conclusão desta dissertação e expõe caminhos para futuros trabalhos.

A seguir, as referências bibliográficas deste trabalho.

E por fim, a referência da produção acadêmica que validou a importância deste trabalho.

1.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo introduz o tema de avaliação de desempenho de descriptografia de vídeo sob demanda utilizando a tecnologia de segurança por *hardware* provida pela Intel, chamado de SGX. A motivação e objetivos geral e específicos da dissertação são apresentados, os trabalhos relacionados e além da exposição das contribuições. No próximo capítulo são apresentados os fundamentos teóricos necessários para o desenvolvimento deste trabalho.

2 FUNDAMENTOS TEÓRICOS

Neste Capítulo são apresentados os conceitos teóricos relevantes para a elaboração deste trabalho: as características de um vídeo digital, as tecnologias de *streaming* de vídeo, conceito e funcionamento de um serviço de Vídeo sob Demanda, estado da arte de criptografia, teoria de criptografia de vídeo, descrição dos parâmetros de desempenho de vídeo e uma abordagem do estado da arte da tecnologia Intel SGX.

2.1 CARACTERÍSTICAS DE UM VÍDEO DIGITAL

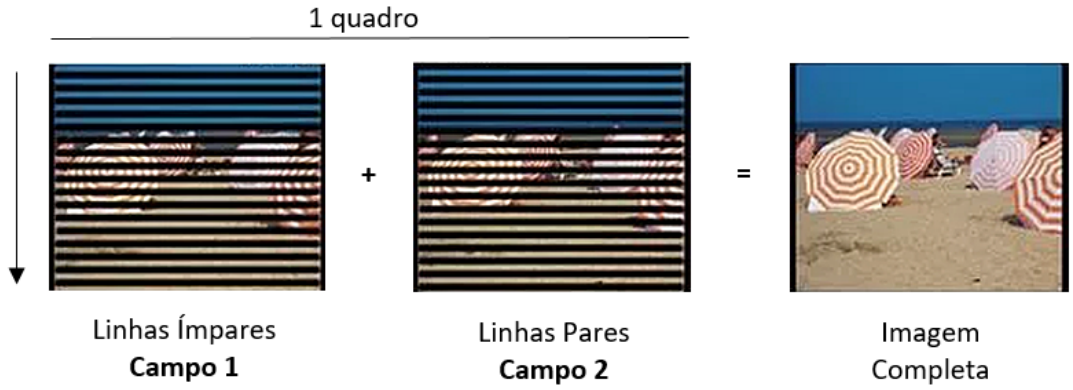
Existem várias características que definem a experiência de visualização do vídeo digital. O cérebro e o olho humano têm maneiras específicas de interpretar imagens em movimento. Algumas das principais características a serem consideradas são: varredura de linha, resolução de vídeo, número de quadros e proporção de tela. Entretanto, essas características podem ser afetadas, em particular, por métodos de compressão de vídeo que são usados nas imagens. As principais características são descritas em detalhes nas próximas subseções (RAMIREZ, 2008):

2.1.1 VARREDURA DE LINHA

A varredura de linha refere-se ao antigo processo de geração de imagem por feixe de elétrons. Há um número predeterminado de linhas possíveis na tela e o aparelho de TV pode varrê-las em ordens diferentes. No entrelaçamento, as linhas horizontais de digitalização de cada quadro são numeradas consecutivamente e divididas em dois campos: os campos par e ímpar, compostos respectivamente pelas linhas pares e ímpares (Figura 1). Ao atualizar as imagens, o sistema atualizará um bloco de campos (meio quadro), o que possibilita transmitir vídeos com largura de banda estreita. A transmissão de apenas metade da imagem reduz a carga geral na mídia de transporte. O olho humano perceberá um nível de mudança e, com a próxima imagem, o movimento será concluído.

Nos sistemas de varredura progressiva, cada transmissão de quadros atualiza todas as linhas, como ilustra a Figura 2. Qualidade e a experiência do espectador é bastante aprimorada mas os requisitos de largura de banda são muito maiores.

Figura 1 - Sistema de Varredura Entrelaçado



Fonte: Adaptado de Pizzotti (2019)

Figura 2 - Sistema de Varredura Progressivo

Fonte: Adaptado de Pizzotti (2019)

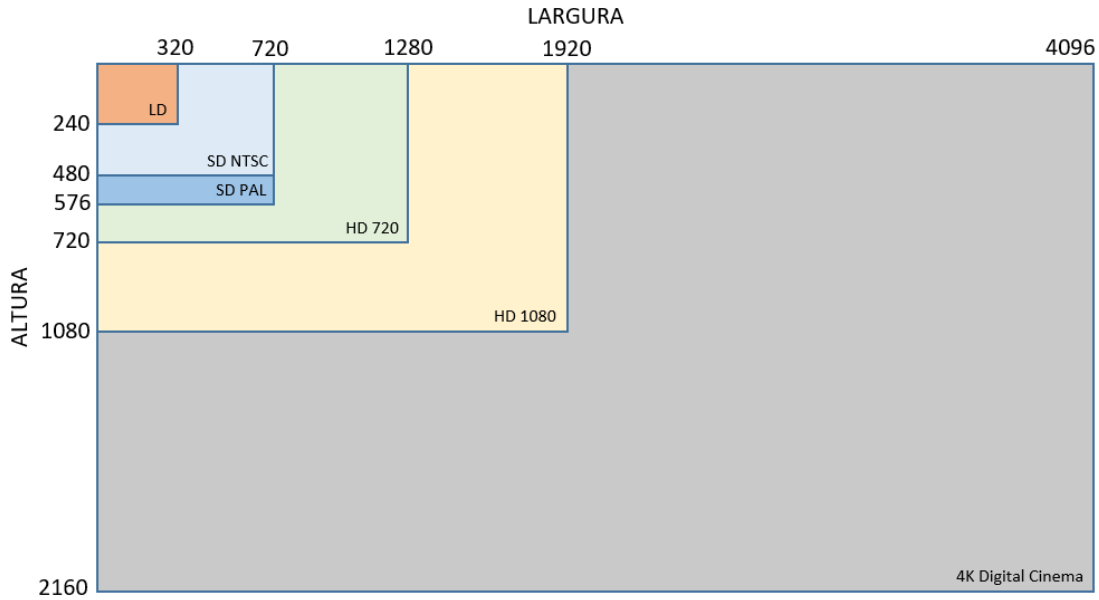
O IPTV requer disponibilidade significativa de largura de banda para uma experiência apropriada do espectador. Em casos na qual a largura de banda é limitada (telefones celulares ou WiMax(PAREIT et al., 2012)), o entrelaçamento pode ser usado. Os assinantes de IPTV esperam uma qualidade de imagem equivalente à terrestre, via satélite e TV a cabo. A digitalização entrelaçada de baixa largura de banda pode não ser aceitável para muitos usuários.

2.1.2 RESOLUÇÃO DE VÍDEO

O mecanismo mais usado para medir a resolução de vídeo é pelo número de *pixels* (Altura vs Largura). O *pixel* ou *Picture Element* é a menor unidade de uma imagem digital.

Entretanto, quanto maior a for resolução, melhor é o detalhe do vídeo. A Figura 3 mostra a diferença entre resoluções em uma região (VISION, 2015).

Figura 3 - Comparação de Resoluções de vídeo em uma região



Fonte: Adaptado de Vision (2015)

O Quadro 1 lista as resoluções mais usadas no mercado e disponibilizadas aos usuários/consumidores/espectadores.

Quadro 1 - Resoluções de Vídeo

Resolução	Largura vs Altura	Descrição
240	320x240	*Low Definition* (LD)
480 NTSC	720x480	*Standard Definition* (SD)
576 PAL/SECAM	720x576	*Standard Definition* (SD)
720	1280x720	*High Definition Ready* (HD Ready)
1080	1920x1080	*High Definition* (HD)
2160	3840x2160	*Ultra High Definition* (UHD/4K)

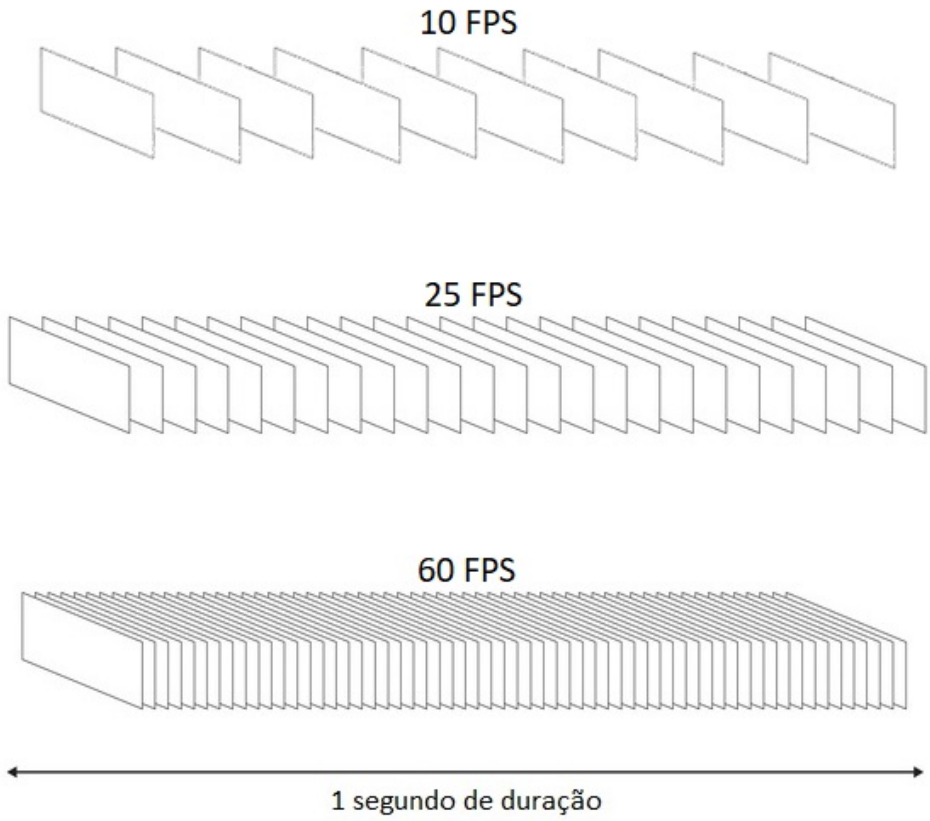
Fonte: Adaptado de Vision (2015)

2.1.3 QUADROS POR SEGUNDO

O número de imagens por segundo mostrado na tela é geralmente chamado de quadros por segundo ou *frames per seconds - fps*. Com um *fps* de 10, é possível criar a ilusão de ótica do movimento: quanto maior o *fps*, melhor será a experiência do usuário/consumidor/espectador. É comum descobrir que cada padrão internacional para vídeo recomenda um *fps* diferente, por exemplo, os padrões PAL e SECAM especificam 25*fps*, enquanto o NTSC especifica 29,97*fps*. O valor do *fps* também tem um impacto nos requisitos de largura de banda para transmissões de IPTV, e também é importante selecionar mecanismos de compactação de vídeo que não levarão

os *fps* a níveis inaceitáveis (RAMIREZ, 2008). A Figura 4 mostra como os números de imagens (10 - 60*fps*) ficam distribuídos em um vídeo de intervalo de 1*s*.

Figura 4 - Ilustração de Quadros por Segundo de 10 a 60*fps*



Fonte: Adaptado de Epiphan (2019)

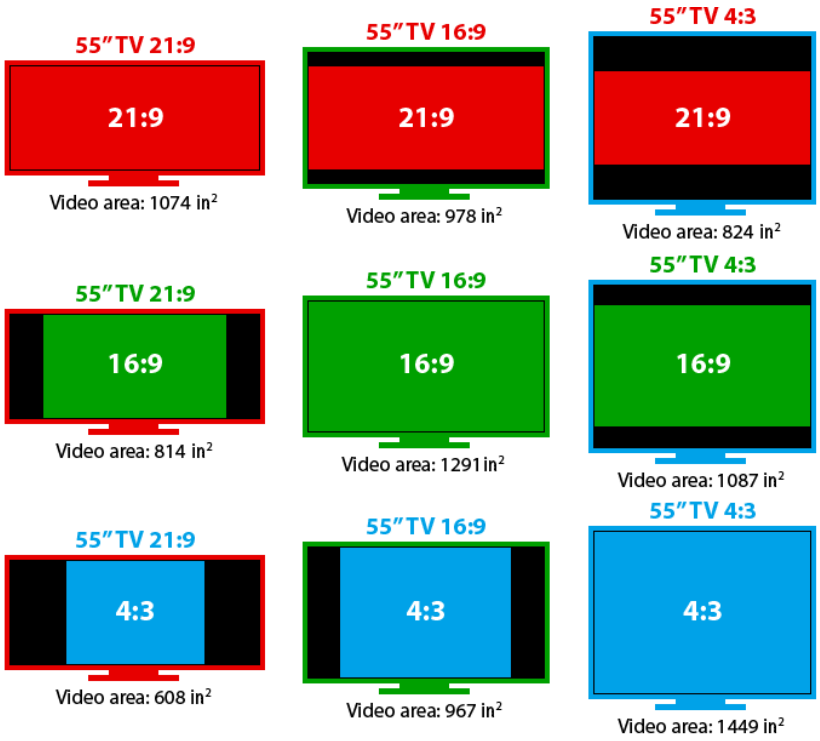
2.1.4 RAZÃO DE ASPECTO

A razão de aspecto é a proporção entre a largura de uma imagem e a altura. As telas de televisão padrão usam 4:3. TVs de alta definição (nova fonte digital) usam uma proporção de 16:9 ou 21:9. Os vídeos podem ser convertidos para diferentes proporções, por exemplo, um vídeo 4:3 (1,33), 16:9 (1,78) e 21:9 (2,33) (RAMIREZ, 2008). A Figura 5 ilustra a razão de aspecto de vídeo aplicado em TVs de 55 polegadas, sendo que as faixas pretas não contém área de vídeo.

2.2 *VIDEO ON DEMAND* (VOD)

O serviço de VOD permite que os usuários selecionem e assistam vídeos de seu interesse. Consequentemente, tais serviços devem permitir que o usuário avance,

Figura 5 - Razões de Aspecto 21:9, 16:9 e 4:3 em uma TV de 55 polegadas



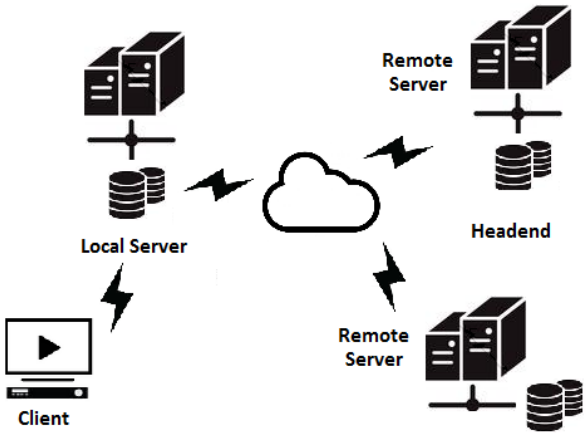
Fonte: Babcock e Demers (2019)

retroceda, suspenda e interrompa o *streaming* de vídeo até a data de expiração definida em um contrato de aquisição do serviço. O conceito básico do VOD é de armazenar canais/conteúdos e encaminhá-los ao usuário/consumidor/espectador. O conteúdo de vídeo (incluindo metadados e outras informações) pode ser mantido em um servidor central de armazenamento e recuperação (servidor de mídia), sendo transmitido simultaneamente para vários usuários/consumidores/espectadores ou replicando em outros servidores pela rede para posterior compartilhamento com os usuários/consumidores/espectadores (IRAWAN, 2013).

Cada acesso a um sistema de VOD requer uma comunicação bidirecional entre o cliente e o servidor, e que cada servidor mantenha um conjunto de vídeos disponíveis para todos os usuários/consumidores/espectadores. O servidor processa solicitações de usuários tão rápido quanto possível, já que centenas ou milhares de usuários/consumidores/espectadores diferentes podem, simultaneamente, requisitar o acesso aos catálogos de filmes ou assistir aos vídeos (LIEBEHERR, 1995). A Figura 6 mostra a arquitetura básica do sistema VOD (KO; KOO, 1996).

O *Headend* é o servidor principal onde fica armazenamento todos os conteúdos de vídeo e onde é também realizado a criptografia dos conteúdos. Os *Remote Servers* e os *Local Servers* são os servidores de *cache* e transmissão de conteúdo de vídeo. E por fim,

Figura 6 - Arquitetura básica de um Sistema de VOD Interativo



Fonte: Ko e Koo (1996)

o equipamento do cliente que tem a função de descodificar e descriptografar todos vídeos recebidos (KO; KOO, 1996).

2.3 TECNOLOGIAS DE *STREAMING* DE VÍDEO

O volume de armazenamento de um servidor de mídia e o tráfego de rede para serviços de VOD aumentam continuamente. O número de vídeos armazenados em um servidor de mídia aumentou drasticamente, juntamente com o crescimento da mídia social e do serviço móvel, e os *streamings* de vídeo precisam ser fornecidos com diferentes qualidades (diferentes taxas de *bits*) à medida que os tipos e recursos dos dispositivos clientes se tornam mais diversificados (MOHAN; LI, 1999). Para o serviço de *streaming* contínuo, os estudos sugeriram ajustar o tráfego de rede selecionando a qualidade de um *streaming* de vídeo de acordo com as condições da rede, otimizando as infraestruturas de distribuição de conteúdo (MOHAN; LI, 1999).

2.3.1 *HTTP LIVE STREAMING* (HLS)

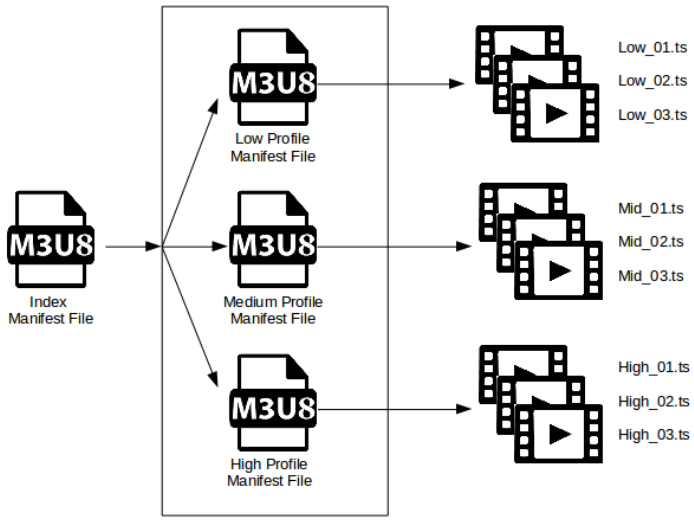
HLS é uma tecnologia de *streaming* adaptativo baseado em HTTP e desenvolvido pela Apple (APPLE, 2016). Ele permite que o STB possa escolher uma versão de qualidade apropriada para cada segmento de vídeo com base na largura de banda disponível entre o servidor e o cliente. O HLS é uma das soluções de *streaming* adaptativo baseado em HTTP e além disso, é o mais difundido no mercado (APPLE, 2016).

Um servidor HLS divide um vídeo em segmentos codificados em uma variedade de resoluções taxas de dados e os armazena em uma unidade de armazenamento. O STB faz o *download* desses segmentos do servidor via HTTP. Em comparação com as tecnologias de

streaming tradicionais, as vantagens do HLS são (CHAKRABORTY et al., 2015):

- Trafegar por *firewall* ou servidor *proxy* que permita o tráfego HTTP padrão;
- Permitir que os segmentos de vídeo sejam armazenados em *cache* por *proxies*, reduzindo assim a carga no servidor de origem e melhorando a velocidade de acesso para *download*;
- Após receber uma solicitação de vídeo do cliente móvel, o servidor da mídia verifica os arquivos de *manifest* que contêm os metadados para cada fluxo codificado e envia segmentos de dados de vídeo em um nível apropriado de taxa de *bits* para o STB. A figura 7 ilustra a composição da estrutura de uma lista de reprodução do HLS.

Figura 7 - Estrutura do *Design* da Lista de Reprodução do HLS



Fonte: Chakraborty et al. (2015)

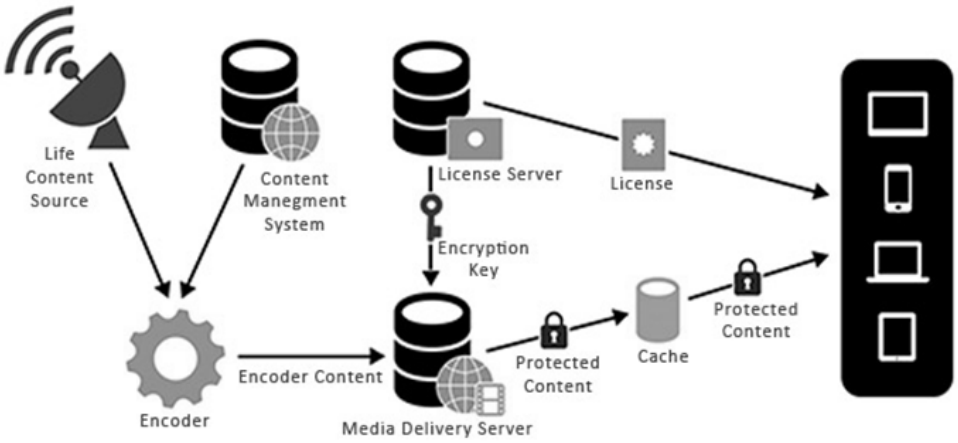
Para a reprodução de vídeo, o STB solicita segmento de vídeo configurados com uma qualidade adequada às condições da rede, consultando os arquivos de *manifest* do servidor. Em seguida, reproduz o segmento de vídeo baixado do servidor. O STB mede o tempo de transmissão do segmento de vídeo solicitado durante o *download* e se ajusta às condições atuais da rede com base no tempo medido e no volume do segmento de vídeo baixado, solicitando automaticamente novos segmentos de vídeo com qualidade adequada (CHAKRABORTY et al., 2015).

2.4 *DIGITAL RIGHTS MANAGEMENT* (DRM)

O gerenciamento de direitos digitais (DRM - *Digital Rights Management*, em inglês) define métodos e tecnologias para controlar o acesso de conteúdo digital de propriedade

de editores, detentores de direitos autorais e/ou indivíduos. Somente usuários autorizados têm acesso a esse conteúdo digital e o conteúdo digital é protegido contra replicação ou transmissão não autorizada. A gestão do acesso à mídia digital é importante para as empresas preocupadas com direitos autorais, particularmente as indústrias de entretenimento, que são parcial ou totalmente dependentes da receita gerada pelo seu trabalho (ENCODING.COM, 2016). A figura 8 ilustra uma arquitetura básica de DRM, desde o processo de codificação do vídeo (*encoder*), passando pelo Servidor de Entrega de Media (*Media Delivery Server*) que juntamente recebe as chaves de criptografia do Servidor de Licença (*License Server*) para finalmente ser disponibilizado o conteúdo de vídeo criptografado ao dispositivo do usuário/consumidor/espectador.

Figura 8 - Arquitetura básica de DRM



Fonte: Chakraborty et al. (2015)

Os dois métodos usuais para integrar DRM em dados de mídia são (ENCODING.COM, 2016):

- Usando um contrato de licença restritivo, no qual o acesso ao material protegido por direitos autorais é concedido de acordo com os contratos legais entre os proprietários do material e os assinantes antes que este possa obter acesso a esse material. Até mesmo o software de domínio público impõe algum acordo legal sobre o material digital fornecido;
- Criptografando, embaralhando e/ou incorporando uma *tag*, no qual o material digital é modificado para bloquear tentativas de copiá-lo ou distribuí-lo para usuários não autorizados e para controlar o acesso a esse material.

Particularmente, o último método envolve a aplicação de criptografia que pode ser baseada em *software* e/ou *hardware*.

2.5 TECNOLOGIAS DE SEGURANÇA PARA *SET-TOP BOXES* (STB)

Ramirez (2008) explica que no mercado de IPTV, existem basicamente três tipos principais de tecnologia usados para proteção de conteúdo:

1. **Sistemas de proteção de conteúdo (CPS)**: O conteúdo é transmitido pelas redes em um servidor criptografado até o STB, para ajudar a proteger contra roubo ou acesso não autorizado;

2. **Sistemas de acesso condicional (CAS)**: Ajuda a garantir que apenas assinantes autorizados tenha acesso ao conteúdo e crie uma proteção contra roubo de serviço;

3. **Gerenciamento de direitos digitais (DRM)**: Gerencia como o conteúdo é usado pelo assinante com base em condições específicas estabelecidas no contrato de distribuição. O termo DRM é aceite no setor para incluir o CPS e opera em conjunto com o *middleware* (Camada de Abstração de *Software* entre uma aplicação e *driver* do *hardware* de um STB) para fornecer serviços relacionados ao CAS. Um servidor DRM suportado por um DRM no decodificador, pode criptografar o conteúdo da fonte e emitir licenças de acesso somente para assinantes autorizados.

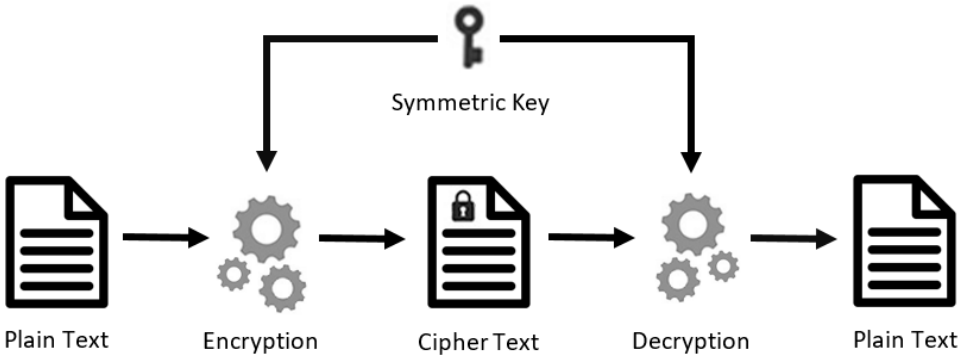
2.6 PRINCÍPIOS DE CRIPTOGRAFIA

A criptografia é uma ciência cujo objetivo é ocultar dados de invasores mal-intencionados, de modo que só possa ser descriptografada por indivíduos autorizados específicos que tenham acesso à uma chave secreta.

Contextualizando, a criptografia pode ser entendida como um conjunto de métodos e técnicas para criptografar/cifrar informações legíveis por meio de um algoritmo de criptografia parametrizado por uma chave, convertendo um texto original em um texto ilegível, denominado texto cifrado, e que posteriormente o receptor autorizado possa descriptografar/decifrar esse texto cifrado e obter a informação original (PIGATTO, 2012; TANENBAUM, 2003).

Existem duas categorias principais de sistemas de criptografia, com diferentes aplicações práticas. A criptografia de chave simétrica (ou criptografia de chave privada) possui este nome porque os processos de criptografia e decriptografia são realizados com uma única chave, ou seja, tanto o emissor quanto o receptor detêm a mesma chave e esta deve ser mantida em segredo para que se possa garantir a confidencialidade das mensagens ou da comunicação, como mostra a figura 9.

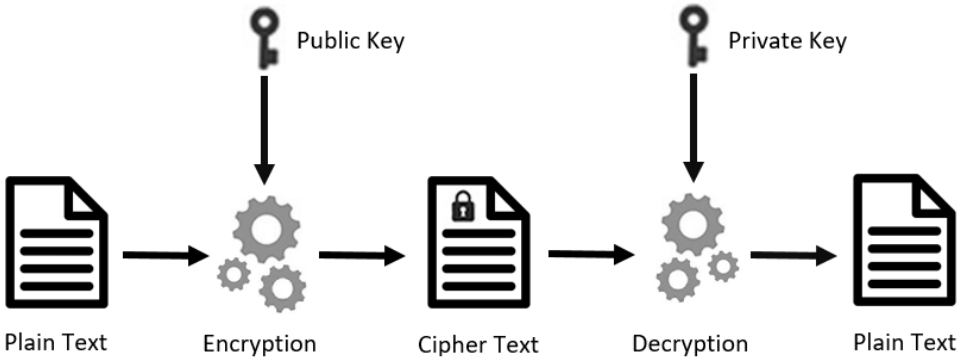
Figura 9 - Processo de Criptografia Simétrica



Fonte: Puri et al. (2004)

E por outro lado, a criptografia assimétrica, mais conhecida como criptografia de chave pública, utiliza um par de chaves denominadas chave privada e chave pública. Qualquer uma das chaves pode ser utilizada para criptografar os dados, porém a mesma não pode ser utilizada para descriptografá-los, isto é, se a criptografia for realizada com a chave pública, somente a respectiva chave privada poderá realizar a descriptografia (PIGATTO, 2012; STALLINGS, 2015). A Figura 10 ilustra esse processo.

Figura 10 - Processo de Criptografia Assimétrica



Fonte: Puri et al. (2004)

(PURI et al., 2004) faz uma comparação entre criptografia assimétrica e criptografia simétrica, a saber:

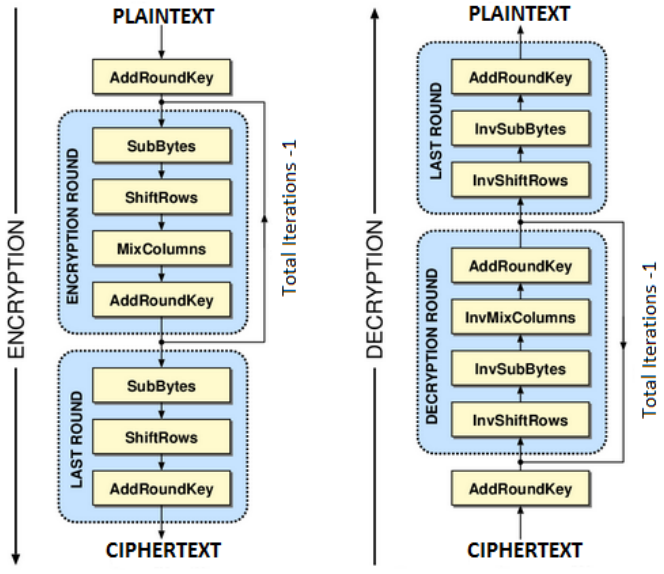
- Assimétrico consome mais tempo e processamento computacional;
- O processamento com chaves simétricas é 200 vezes mais rápido do que o equivalente com chaves assimétricas;
- Como a criptografia assimétrica emprega a fatoração de grandes números, a derivação de duas chaves não é adequada para aplicações em tempo real.

Portanto, os algoritmos de chave simétrica são os mais escolhidos, em vez dos algoritmos de chave assimétrica, afirma (PURI et al., 2004). Segundo Othman et al. (2012), o algoritmo AES é a melhor criptografia de chave simétrica para transmissão de vídeo em tempo real, em termos de segurança e tempo. O AES é um dos algoritmos criptográficos mais fortes conhecidos até o momento, pois baseia-se na rede de substituição-permutação e é rápido tanto em *hardware* quanto em *software*.

2.6.1 ADVANCED ENCRYPTION STANDARD (AES)

O algoritmo AES é o padrão de criptografia adotado pelo NIST (*National Institute of Standards and Technology*) para proteger os dados durante a comunicação. O AES possui um tamanho de bloco de dados fixo de 128 *bits* (ou 16 *bytes*), além de um tamanho de chave de: 128, 192 ou 256 *bits*. Os blocos de dados 16 *bytes* são mapeados em matrizes 4x4 chamados de *State*. E todas as operações internas do algoritmo do AES são realizadas nessas matrizes. O algoritmo AES é um algoritmo de chave simétrica e também é iterativo. As iterações são chamadas de *rounds* e total do número de iterações ou *rounds* são 10, 12 ou 14 para tamanhos de chaves de 128, 192 e 256 *bits*, respectivamente (STALLINGS, 2015).

Figura 11 - Arquitetura do algoritmo AES



Fonte: Arrag et al. (2014)

Na figura 11 mostra a arquitetura em blocos no processo de criptografia e descriptografia do algoritmo AES. Na criptografia do algoritmo do AES, cada *round* (com exceção do último *round*) consiste em quatro transformações: o *SubBytes()*, o *ShiftRows()*, o *MixColumns()* e o *AddRoundKey()*. O último *round* não tem a transformação *MixColumns()*. E

na descriptografia, consiste as seguintes transformações inversas: *InvShiftRows()*, *InvSubBytes()* e *InvMixColumns()* (ARRAG et al., 2014) e (STALLINGS, 2015).

2.7 CRIPTOGRAFIA DE VÍDEO

Segundo Shah e Saxena (2011), há uma classificação de algoritmos voltados para criptografia de video, além de vários parâmetros de desempenho para comparação desses algoritmos. A classificação, por sua vez, é baseada em um único sentido, isto é, na criptografia de dados apenas.

2.7.1 CLASSIFICAÇÃO DE ALGORITMOS

Conforme Shah e Saxena (2011), os algoritmos voltados para criptografia de vídeo, são classificados basicamente em quatro categorias, a saber:

1. **Criptografia Total**: O vídeo é comprimido primeiro e depois criptografado usando algoritmos tradicionais de criptografia, como o AES. Essa técnica não é a mais adequada para aplicações de *streaming* de vídeo em tempo real, devido ao alto processamento de computação e por ter uma baixa velocidade;

2. **Criptografia baseada em Permutação**: Os algoritmos dessa categoria usam diversas permutações com chave secreta para criptografar ou embaralhar o conteúdo de vídeo. Além disso, não é necessário embaralhar cada *byte*;

3. **Criptografia Seletiva**: Os algoritmos dessa categoria criptografam randomicamente alguns *bytes* de um *frame* de vídeo e consequentemente, reduz a complexidade computacional;

4. **Criptografia Perceptiva**: Os algoritmos dessa categoria trabalham com um fator P que é usado para ajustar degradação do vídeo após a aplicação de criptografia nos *frames* de vídeo. Esse ajuste é baseado em avaliações para Qualidade de Vídeo.

2.7.2 PARÂMETROS DE DESEMPENHO

Shah e Saxena (2011) explicam que existem basicamente cinco parâmetros de desempenho para avaliação e comparação de algoritmos para criptografia de video, a saber:

1. **Taxa de Criptografia**: Este parâmetro mede a taxa entre o tamanho da parte criptografada e o tamanho total dos dados. A taxa de criptografia deve ser minimizada para reduzir complexidade computacional;

2. **Velocidade**: Em muitos aplicativos de vídeo em tempo real, é importante que os algoritmos de criptografia e descriptografia sejam rápidos o suficiente para atender aos requisitos em tempo real;

3. **Compressão Amigável**: Um algoritmo de critografia é considerado amigável à compactação se tiver pouco ou nenhum impacto na eficiência da compactação de dados, ou seja, não introduz muitos dados adicionais que sejam necessários para descriptografia;

4. **Conformidade de Formato**: O fluxo de *bits* criptografados deve ser compatível com o *decoder* de vídeo, isto é, o *decoder* deve ser capaz de descodificar o fluxo de *bits* criptografado sem descriptografia;

5. **Segurança Criptográfica**: A definição de segurança criptográfica é quando um algoritmo de criptografia é muito seguro contra ataques de força bruta e de diferentes tipos de ataques.

2.8 INTEL *SOFTWARE GUARD EXTENSIONS* (SGX)

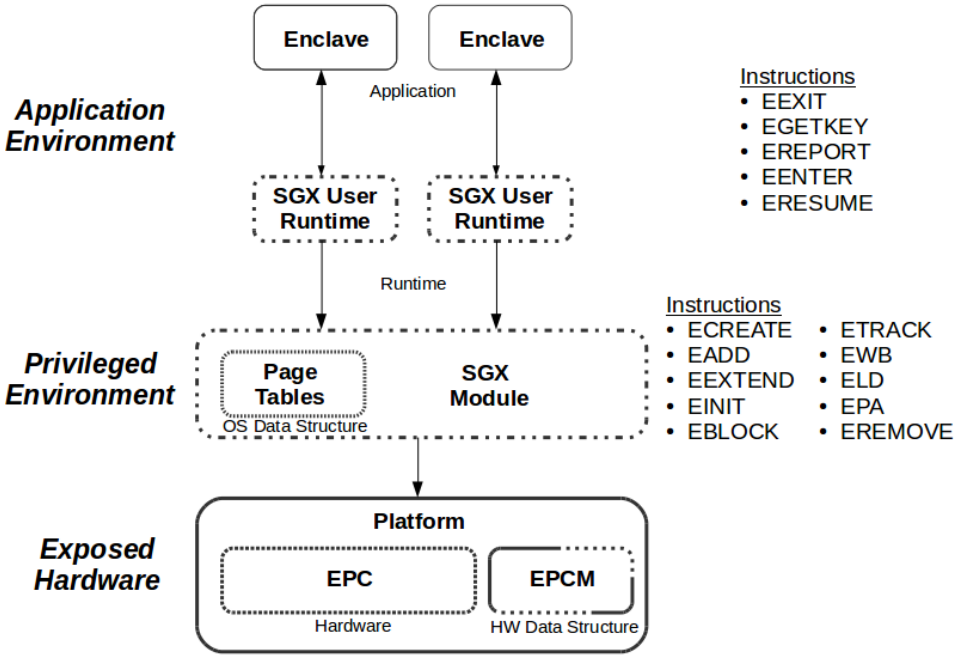
O Intel SGX (*Software Guard Extensions*) foi implementado com o Skylake em outubro de 2015 (KNIGHT, 2015). Ele é um conjunto de extensões x86-64ISA e é um recurso de ativação ativado pela BIOS que permite configurar ambientes de execução protegidos (denominados *enclaves*) sem exigir confiança em nada além do processador e do código que os usuários colocam dentro de seus *enclaves* (ANATI et al., 2013).

O SGX reduz a superfície de ataque proibindo o acesso a *software* com privilégios maiores, como *drivers* de sistema ou *kernel*, *hipervisors*, manipuladores de SMM (modo de gerenciamento de sistema) etc. Além disso, as *enclaves* são criptografados enquanto residem na memória do sistema protegidos de ataques de varredura ou de ataques físicos. O SGX funciona fornecendo várias bordas de proteção em torno de códigos e/ou dados sensíveis. Uma visão geral de alto nível do panorama do SGX é mostrada na Figura 12.

2.8.1 *ENCLAVE*

Ao carregar um *enclave* na memória, a CPU (*Central Processor Unit*) mede seu conteúdo em um *log* de *hash* criptografado encadeado e o armazena em um registrador chamado

Figura 12 - Arquitetura em Alto nível do *Hardware/Software* do SGX



Fonte: Intel (2016)

`MRENCLAVE`, semelhante ao PCR (*Platform Configuration Register*) e os TPMs (*Trusted Platform Modules*). Antes de executar uma *enclave*, a CPU verifica o conteúdo de `MRENCLAVE` em relação a uma versão assinada pelo fornecedor e aborta em caso de uma incompatibilidade. Portanto, a CPU garante a integridade da inicialização da *enclave* (TCG, 2014).

Para avaliar a segurança da *enclave*, o SGX fornece mecanismos de atestação (ANATI et al., 2013). A atestação local permite que uma *enclave* verifique a confiabilidade de outra *enclave* executado na mesma CPU física. O atestado remoto pode ser usado por uma parte remota para verificar a confiabilidade de um *enclave* executado em uma CPU Intel genuína. Ele também permite o provisionamento inicial de chaves para definir a comunicação entre as *enclaves*: isso é necessário, pois o código da *enclave* é público, não permitindo a incorporação de segredos diretamente em seu código binário. O atestado é baseado em uma estrutura de relatório assinada e gerada pela CPU que pode conter dados adicionais do usuário (incluindo chaves e segredos, como mencionado anteriormente).

O SGX também permite que uma *enclave* obtenha uma chave de vedação limitada à CPU local e ao criador da *enclave*. Portanto, a mesma chave de vedação só pode ser consultada unicamente pela *enclave* que a criou, se carregada corretamente na mesma CPU genuína da Intel. A *enclave* pode usar a chave de vedação para criptografar e descriptografar dados arbitrários para armazenamento *off-line*, preservando os dados em várias execuções de uma

determinada *enclave* (INTEL, 2016).

Durante a inicialização da *enclave*, a CPU verifica a *enclave* `EINITTOKEN` (vários atributos a serem aplicados, incluindo o sinalizador de modo de depuração). Ele deve ser assinado por uma chave de lançamento especial, que é de propriedade das *enclaves* de lançamento emitidos pela Intel. Portanto, ao emitir *enclaves* de inicialização apropriados, a Intel controla quais enclaves devem ser executados no modo de depuração ou produção. O SDK de avaliação do SGX é fornecido com um enclave de inicialização apenas para *enclaves* de depuração (INTEL, 2017a), enquanto as *enclaves* para o modo de produção requerem licenças apropriadas da Intel (JOHNSON et al., 2016).

2.9 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo abordou os fundamentos teóricos de embasamento para a elaboração dessa dissertação. Foram apresentados: as características de um vídeo digital, as tecnologias de *streaming* de vídeo, conceito e funcionamento de um serviço de Vídeo sob Demanda (VOD), estado da arte de criptografia, teoria de criptografia de vídeo, descrição dos parâmetros de desempenho de vídeo e uma abordagem do estado da arte da tecnologia Intel SGX. O próximo capítulo explica a metodologia a ser aplicada.

3 METODOLOGIA

Este capítulo detalha a metodologia aplicada neste trabalho: o *design* do sistema, os procedimentos para avaliação de desempenho e detalhes da implementação, incluindo o processo de descriptografia do vídeo na *enclave*, a forma de medição de tempo para avaliação de desempenho, configurações de *hardware* e *software* e descrição das amostras de vídeo usadas na avaliação.

3.1 *HARDWARE*

Duas configurações de computador são usadas para avaliar o desempenho de descriptografia de vídeo, ambos com suporte ao SGX: a primeira consiste em um PC com um processador Intel® Xeon E3-1280 v6 4-core de 3.9GHz e 24GB de RAM. A última configuração consiste em um notebook Dell Latitude 5000 Series (5480) com um processador Intel® Core I7-7600U v6 de 4 núcleos de 2.6GHz e 16GB de RAM. O tamanho da memória da *enclave* foi fixada em 128MB diretamente na configuração da BIOS.

3.2 SISTEMA OPERACIONAL, SDK E *TOOLCHAIN*

Ambos computadores utilizam o sistema operacional Ubuntu 16.04 64bits LTS (CANONICAL, 2019) com o *kernel* versão 4.4.0-121-generic. O SDK do Intel SGX instalado, foi a versão 2.0.40950. E para o funcionamento do referido SDK, foi necessário instalar o SGX *Device Driver* para o Sistema Operacional Linux na versão 2.0, com o objetivo de prover o acesso aos registradores e a *enclave* do processador em questão. Antes da instalação do SDK e do *Device Driver* do Intel® SGX, os códigos-fonte foram compilados para o processador citado, utilizando o compilador GCC - (*GNU Compiler Collection)* (GNU, 2019) na versão 5.4.0.

3.3 AMOSTRAS DE VÍDEO, SEGMENTOS DE VÍDEOS E TAMANHOS DE *BUFFER* DE ENTRADA

O Quadro 2 lista as amostras de vídeos de referência com seus respectivos parâmetros (XIPH.ORG, 2016). E seguindo os procedimentos atuais de VOD, usou-se a ferramenta

`ffmpeg` (FFMPEG, 2018) para particionar todos os vídeos em segmentos de vídeo de dois segundos resultando em um total de cinco segmentos de vídeo. Considerou-se tal tamanho como ideal para uma configuração de rede de cenários reais com conexões persistentes de servidores da Web/Conteúdos e HTTP 1.1, incluindo também a transmissão de vídeo ao vivo (LEDERER, 2015). Junto com esses segmentos de vídeo, foi produzido um arquivo de configuração de índice (extensão `m3u8`) para cada vídeo (em aplicativos reais, todos os arquivos são transportados do servidor para STBs em redes privadas). Apesar do HLS suportar várias resoluções de vídeo que podem ser definidas em um mesmo arquivo de configuração de índice (*manifest*), neste trabalho foi considerado apenas uma única resolução.

Quadro 2 - Amostras de Vídeo

Nome do Vídeo	Resolução	Proporção	Quadros	Duração (seg.)
Coast Guard	720x480p	4:3	300	10
Foreman	720x480p	4:3	300	10
Ducks Take Off	1280x720p	16:9	500	10
In To Tree	1280x720p	16:9	500	10
Ducks Take Off	1920x1080p	16:9	500	10
In To Tree	1920x1080p	16:9	500	10
Old Town Cross	3840x2160p	16:9	500	10
Park Joy	3840x2160p	16:9	500	10

Fonte: Xiph.Org (2016)

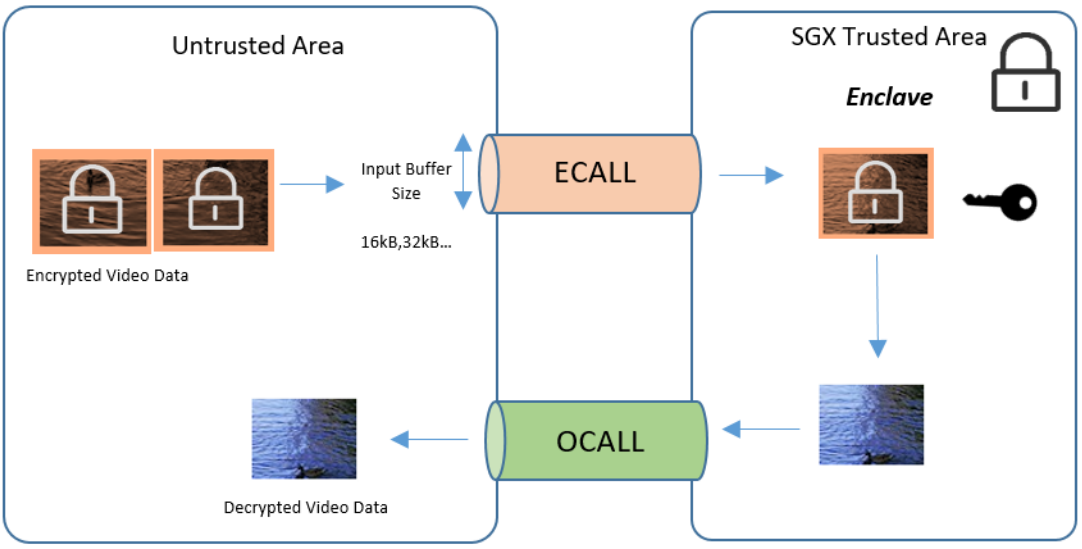
Os segmentos de vídeo de resoluções 480p, 720p e 1080p foram recodificados para H.264 (WATKINSON, 2008) e o de resolução 2160p foi recodificado para H.265 (WIEN, 2015) e todos no modo *Baseline Profile*. Esse modo foi projetado principalmente para aplicativos de baixo custo com recursos de computação limitados e amplamente utilizados em videoconferência e aplicativos móveis, segundo afirma Huang (2008). Um STB com SGX deve descriptografar um segmento de vídeo inteiro em menos de quatro segundos, sem comprometer a qualidade na recepção (LEDERER, 2015).

Embora o tamanho dos segmentos de vídeo sejam fixos no tempo, esses arquivos têm tamanhos diferentes em *bytes*. Essa variação está diretamente relacionada com a codificação do conteúdo do vídeo (principalmente a distribuição *inter* e *intra-frames* (WATKINSON, 2008) de *pixels* coloridos).

Foram usados tamanhos de *buffer* de entrada de dados fixos para transferir dados de tamanho fixo de/para a função de descriptografia dentro de *enclaves* SGX (tamanho do *buffer* $\leq$ tamanho do segmento). Isso foi feito para obter o tamanho ideal do *buffer* ao usar o SGX para descriptografar o conteúdo do vídeo. O tamanho desses *buffers* de entrada de dados variam de

1kB a 1024kB. A Figura 13 ilustra o processo de descriptografia de um vídeo entrando dados de vídeo criptografado através do ECALL e saindo o dado de vídeo descriptografado em uma OCALL, pertencente a uma *enclave*.

Figura 13 - Processo de descriptografia de vídeo ECALL/OCALL



Fonte: Autoria Própria

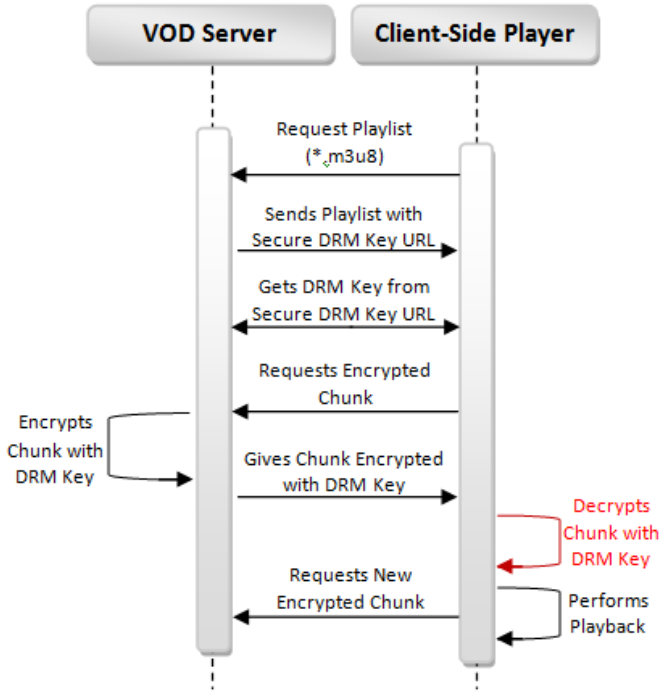
3.4 PROPOSTA DA SOLUÇÃO

Normalmente, um STB precisa recuperar os segmentos de vídeo criptografados (blocos de dados de vídeo) de um servidor de mídia, descriptografá-los dentro de *enclaves* e, finalmente, reproduzí-los. Considerando que o paradigma SGX que impõe atrasos adicionais a esse procedimento devido à necessidade de transferir dados de/para *enclaves*, avalia-se a viabilidade de usar STBs baseados em SGX para aplicativos VOD. Particularmente, como o vídeo é dividido em blocos de dados quando transferidos de um servidor de mídia para um STB, diferentes tamanhos de blocos de dados são avaliados para determinar o tamanho ideal para o processo de VOD. Por isso, concentrou-se em medir o desempenho da descriptografia dentro de uma *enclave*. A Figura 14 mostra um diagrama de sequência básico entre o servidor VOD e um *player* de vídeo presente em um STB (cliente). O item destacado em vermelho na Figura 14 enfatiza o foco da pesquisa.

3.5 ALGORITMOS DE CRIPTOGRAFIA E DETALHES DE IMPLEMENTAÇÃO

Foi usado a codificação AES-GCM de 128 *bits* para descriptografar segmentos de vídeo e foi avaliado o desempenho desse processo usando *enclaves* SGX e a biblioteca

Figura 14 - Diagrama de sequência básico de comunicação VOD Cliente-Servidor



Fonte: Costa et al. (2018)

OPENSSL, para área não segura. O algoritmo AES-GCM foi selecionado pelo fato de sua implementação já seja fornecida para o SDK do SGX. Por outro lado, o OPENSSL foi escolhido porque é uma das bibliotecas *open source* mais utilizadas, bem documentada e integrada numa ampla gama de aplicações (BUTIN et al., 2017). Assim, tem-se duas soluções STB para descriptografar os segmentos de vídeo com a cifra AES-GCM: usando OPENSSL (ou solução não SGX) e usando SGX (rodando dentro de *enclaves*). Para implementar a solução de *enclave* SGX, usou-se a biblioteca `libsgx_urts` oferecida pela Intel para criar e destruir *enclaves*. Um `ECALL` para uma rotina dentro do SGX é chamado toda vez que um segmento de vídeo precisa ser descriptografado, transferindo o próprio *buffer* de dados para a *enclave*. Da biblioteca de algoritmos de criptografia SGX (`sgx_tcrypto`), foi usado a função `sgx_rijndael128GCM_decrypt()` para descriptografia de partes. Uma variável de tipo de chave (`sgx_aes_gcm_128bit_key_t`) foi estaticamente definida dentro da *enclave* para conter a chave secreta (nenhum tratamento para distribuição de chaves foi realizado).

Na solução OPENSSL, foi usado a biblioteca `Crypto` oferecida pelo OPENSSL para implementar a descriptografia dos segmentos de vídeo. O equivalente à solução SGX, foi usado as funções de descriptografia de alto nível (OPENSSL-EVP) `EVP_aes_128_gcm()` configurando o algoritmo do AES-GCM. Embora haja um atraso envolvido na transferência de dados de/para bibliotecas OPENSSL, não há mecanismo de criptografia de memória, pois o

código binário final é executado fora das *enclaves*.

3.6 PROCEDIMENTO DE AVALIAÇÃO DE DESEMPENHO

O procedimento de avaliação de desempenho proposto por Jain (1991), permite determinar se um fator tem um efeito significativo ou se a diferença observada é simplesmente devido a variações aleatórias causadas por erros de medição e parâmetros que não foram controlados. E na etapa de projeto e análise de experimentos, alguns termos são utilizados:

- **Variável de resposta**: é a saída de um experimento, normalmente a métrica utilizada para avaliar o desempenho do sistema;
- **Fatores**: são as variáveis que afetam a variável de resposta do sistema;
- **Níveis**: são os valores que um determinado fator pode assumir;
- **Interação**: indica a dependência entre os fatores avaliados.

O procedimento de avaliação de desempenho proposta por Jain (1991), é composta das seguintes equações matemáticas: desvio padrão, variância, intervalo de confiança e 2^K Fatorial Completo, que serão mostrados nas subseções a seguir.

3.6.1 DESVIO PADRÃO E VARIÂNCIA

O desvio padrão e a variância são parâmetros que medem a dispersão de um conjunto de dados em relação à média. A variância, representada por σ^2, é definida como o desvio quadrático médio da média e é calculada sobre uma amostra de dados partindo da Equação 1, onde n corresponde ao número de elementos da amostra coletada e $(x_i - \bar{x})^2$ corresponde ao quadrado da distância entre uma amostra x_i e a média da amostra $\bar{x}$. O quadrado da diferença é utilizado para computar o valor absoluto da distância.

$$\sigma^2 = \sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n-1} \tag{1}$$

O desvio padrão, representado por σ, refere-se à raiz quadrada da variância (Equação 2). Tem a mesma função da variância, porém apresenta a vantagem de permitir uma interpretação direta da variação da amostra de dados, pois o desvio padrão é expresso na mesma unidade de medida dos dados.

$$\sigma = \sqrt{variancia} = \sqrt{\sigma^2} \tag{2}$$

3.6.2 INTERVALO DE CONFIANÇA

Para encontrar uma estimativa perfeita para a média seria necessário utilizar um número infinito de amostras. Como isso não é possível, adota-se a obtenção de limites probabilísticos (intervalo de confiança), c_1 e c_2, de modo que a média exata $\bar{x}$ pertença ao intervalo $[c_1, c_2]$, com uma certa probabilidade $(1-\alpha)$ de acerto. A Equação 3 representa a probabilidade de um intervalo de confiança estar correto, onde α corresponde ao nível de significância e $1-\alpha$ corresponde ao coeficiente de confiança.

$$P(c_1 \leq \bar{x} \leq c_2) = 1 - \alpha \tag{3}$$

De modo geral, o intervalo de confiança é explicitado em um percentual próximo a 100%, por exemplo, 90% ou 95%. Já o nível de significância α é explicitado como fração e é usualmente próximo de zero, por exemplo, 0,05 ou 0,1 (KAMIENSKI et al., 2002).

3.6.3 2^K FATORIAL COMPLETO

O fatorial completo descreve a influência de cada um dos fatores elencados com relação a um parâmetro de resposta, de acordo com os níveis escolhidos. É calculado com base em um modelo de regressão linear da forma como mostra a Equação 4 (exemplo com 2 fatores, ou $K = 2$).

$$y = q_0 + q_A x_A + q_B x_B + q_{AB} x_A x_B \tag{4}$$

na qual y é a variável de resposta e x_w (w é o fator, nesse caso A e B) pode assumir os valores **-1** e **1** que representa cada um dos 2 níveis escolhidos. Combinando os níveis de cada fator, obtêm-se $2^{k=2}$ cenários (Equação 5):

$$\begin{aligned} y_1 &= q_0 - q_A - q_B + q_{AB} \\ y_2 &= q_0 + q_A - q_B - q_{AB} \\ y_3 &= q_0 - q_A + q_B - q_{AB} \\ y_4 &= q_0 + q_A + q_B + q_{AB} \end{aligned} \tag{5}$$

Deste modo, pode-se obter os valores de q_w em relação à variável de resposta de cada

um dos cenários, como observado nas Equações 6:

$$\begin{aligned} q_0 &= \tfrac{1}{4}(y_1 + y_2 + y_3 + y_4) \\ q_A &= \tfrac{1}{4}(-y_1 + y_2 - y_3 + y_4) \\ q_B &= \tfrac{1}{4}(-y_1 - y_2 + y_3 + y_4) \\ q_{AB} &= \tfrac{1}{4}(y_1 - y_2 - y_3 + y_4) \end{aligned} \tag{6}$$

A importância de um fator é medida pela proporção na variação total da variável de resposta que é influenciada por esse fator. Se um fator é responsável por 95% e outro fator por 5% da variação total, então o segundo fator pode ser considerado não importante em muitas situações práticas. O dividendo da Equação 1 é chamado de variação total de y ou soma dos quadrados total (*Sum of Squares Total*), como mostra a Equação 7, onde K é a quantidade de fatores, y_i é a variável de resposta para o experimento i e $\bar{y}$ é a média da variável de resposta de todos os experimentos (JAIN, 1991).

$$\text{Variação total de } y = \text{SST} = \sum_{i=1}^{2^K} (y_i - \bar{y})^2 \tag{7}$$

Para $k = 2$, a variação pode ser dividida em três partes, segundo a Equação 8.

$$SST = 2^2 q_A^2 + 2^2 q_B^2 + 2^2 q_{AB}^2 \tag{8}$$

As três partes do lado direito da equação representam a porção da variação total explicada pelos fatores A, B e pela interação AB, respectivamente. Então, $2^2 q_A^2$ é a porção de SST explicada pelo fator A e é denotado por SSA. Da mesma forma, SSB é $2^2 q_B^2$ e SSAB é $2^2 q_{AB}^2$. Assim temos a Equação 9.

$$SST = SSA + SSB + SSAB \tag{9}$$

A fração explicada pelo fator A pode ser representada pela Equação 10. O mesmo se aplica ao fator B e à interação AB.

$$\text{Fração da variação explicada por } A = \frac{SSA}{SST} \tag{10}$$

Quando expressada em porcentagem, essa fração mede a importância do fator A. Os fatores com alta porcentagem de variação são considerados importantes.

3.7 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo mostrou a metodologia aplicada neste trabalho: o *design* do sistema, incluindo o processo de descriptografia do vídeo na *enclave*, configurações de *hardware* e *software* para o ambiente de testes, descrição das amostras de vídeo usadas na avaliação e os procedimentos para o cálculo de avaliação de desempenho proposta por Jain (1991). O próximo capítulo mostra os resultados obtidos utilizando o SGX e o OPENSSL em quatro resoluções de vídeo (480p, 720p, 1080p e 2160p). Cada resolução de vídeo consiste em um estudo de caso para abordar em detalhes os testes, resultados e análises.

4 RESULTADOS E DISCUSSÕES

Este capítulo descreve os resultados obtidos com a implementação de um algoritmo para coletar informações de desempenho de descriptografia de vídeo dentro de uma *enclave* SGX e também de uma biblioteca chamada OPENSSL, utilizando resoluções distintas de vídeos (480p, 720p, 1080p e 2160p), e a comparação entre o algoritmo de criptografia AES-GCM implementado tanto no SDK do Intel SGX quanto na biblioteca OPENSSL e em diversos tamanhos de *buffers* de entrada (1kB, 2kB, 4kB, 8kB, 16kB, 32kB, 64kB, 128kB, 256kB, 512kB e 1024kB).

4.1 ANÁLISES DOS RESULTADOS

A avaliação de desempenho do processo de descriptografia de vídeo foi aplicada a todos os vídeos listados na Tabela 2. Entretanto, de todos os vídeos avaliados, apenas quatro vídeos (Figura 15), um de cada resolução (480p, 720p, 1080p e 2160p), são escolhidos e analisados nos estudos de caso deste trabalho.

O uso da tecnologia SGX para proteger os segmentos de vídeo inevitavelmente introduz-se *overheads*. Os resultados indicam que esse *overhead* é de mesma natureza daquela reportada por (GJERDRUM et al., 2017) e (HARNIK, 2017), ou seja, decorre de:

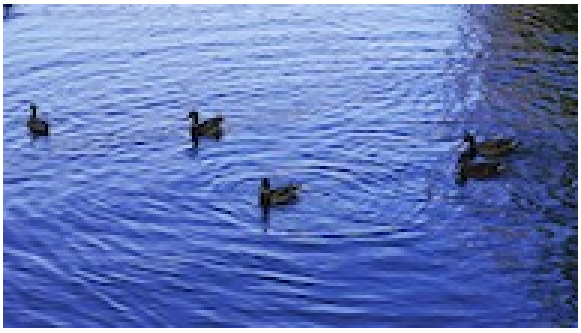
1. **gerenciamento de *enclaves***: principalmente durante a criação e destruição dessas *enclaves*;
2. **ECALLs/OCALLs**: exige muito do processador durante a preparação de mudança de contexto de áreas não confiáveis para *enclaves*.

Aqui são analisados os resultados da comparação entre o algoritmo de criptografia AES-GCM implementado tanto no SDK do Intel SGX quanto na biblioteca OPENSSL (sem usar *enclaves*) na operação de descriptografia. Como mencionado, variou-se o tamanho dos *buffers* de entrada (1kB, 2kB, 4kB, 8kB, 16kB, 32kB, 64kB, 128kB, 256kB, 512kB e 1024kB), que são usados para enviar e receber dados da *enclave*.

Neste trabalho visou-se obter a máxima informação com o número mínimo experimentos, utilizando o fatorial completo. Os resultados mostrados são a média das

Figura 15 - Vídeos utilizados nos estudos de caso

(a) *Coast Guard* 4:3



(b) *Ducks take off* 16:9

(c) *In to tree* 16:9

(d) *Old town cross* 16:9

Fonte:Xiph.Org (2016)

quantidades medidas (100 repetições/experimento) e o intervalo de confiança com um nível de confiança de 95%, conforme é definido por (JAIN, 1991).

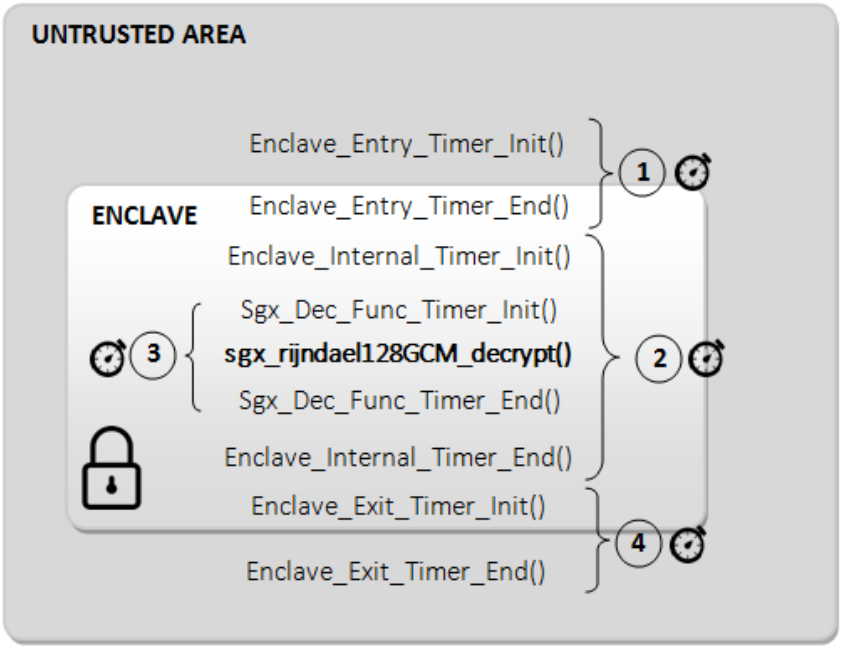
Além disso, analisou-se a influência de três fatores importantes:

1. **Fator A**: consiste na biblioteca SGX e OPENSSL;
2. **Fator B**: consiste no tamanho dos *buffers* de entrada de dados;
3. **Fator C**: representa o tamanho dos arquivos dos segmentos de vídeo.

O processo completo envolve etapas de movimentação de dados de uma área não confiável para a *enclave*, descriptografia desses dados e sua movimentação reversa para área não confiável (para projeção efetiva do segmentos de vídeo). Visando uma avaliação detalhada do impacto de cada etapa no processo geral, a medição dos tempos foram realizados de acordo com a Figura 16, conforme a descrição das etapas a saber:

- **Etapa 1**: representa o tempo gasto para entrar na área segura (*enclave*);
- **Etapa 2**: é identificada como “Outros” por representar as operações executadas dentro da *enclave* EXCETO a operação de descriptografia;
- **Etapa 3**: é o próprio tempo de descriptografia, isto é, utilizando a função de descriptografia em questão;
- **Etapa 4**: é o tempo gasto para enviar dados de volta para a área não confiável.

Figura 16 - Contagem de tempo no processo de descriptografia de vídeo



Fonte: Costa et al. (2018)

Todas as medições realizadas nesta análise estão em microsegundos, devido o tempo de descriptografia ter uma granularidade muito pequena.

Para atender os fatores no cálculo da medição de desempenho, dois segmentos de vídeo são considerados:

1. Primeiro segmento de vídeo - contém os parâmetros de vídeo e informações dos segmentos de vídeo subsequentes;
2. Segundo segmento de vídeo - contém dados de vídeo.

A seguir, são analisados os Estudos de Caso em questão.

4.1.1 ESTUDO DE CASO 1: VÍDEO DE RESOLUÇÃO 480P (*COAST GUARD*)

Este primeiro estudo de caso, é analisado o vídeo de resolução de 720x480 de entrelaçamento progressivo, chamado *Coast Guard*.

Para iniciar a análise dos resultados deste primeiro estudo de caso, a Figura 17 mostra um amplo panorama de desempenho de diferentes tamanhos de *buffer* de entrada de dados, na transferência de dados de/para a função de descriptografia dentro da *enclave*.

Observa-se que de todos os tamanhos de *buffer* de entrada de dados, o de 32kB oferece o melhor desempenho, em relação aos demais, considerando o tempo gasto relacionado à execução de várias transferências de dados em pequenas partes para a *enclave*.

Observa-se na Figura 17, apenas com resultados do SGX, que:

- O desempenho melhora a medida que o tamanho dos *buffers* de entrada de dados aumenta de 1kB à 32kB, já que há uma redução no tempo de processamento;
- O desempenho piora a medida que o tamanho dos *buffers* de entrada de dados aumenta de 32kB para 64kB. E a partir de 64kB até chegar em 1024kB, há uma pequena oscilação no tempo de processamento.

Além do mais, quanto maior fosse o tamanho dos *buffers* de entrada de dados, melhor seria o seu desempenho, isto é, haveria menos atividades de ECALL/OCALL. Mas não foi o que ocorreu nas medições de desempenho comparando o tamanho dos *buffers* de entrada de dados de 32kB e 64kB.

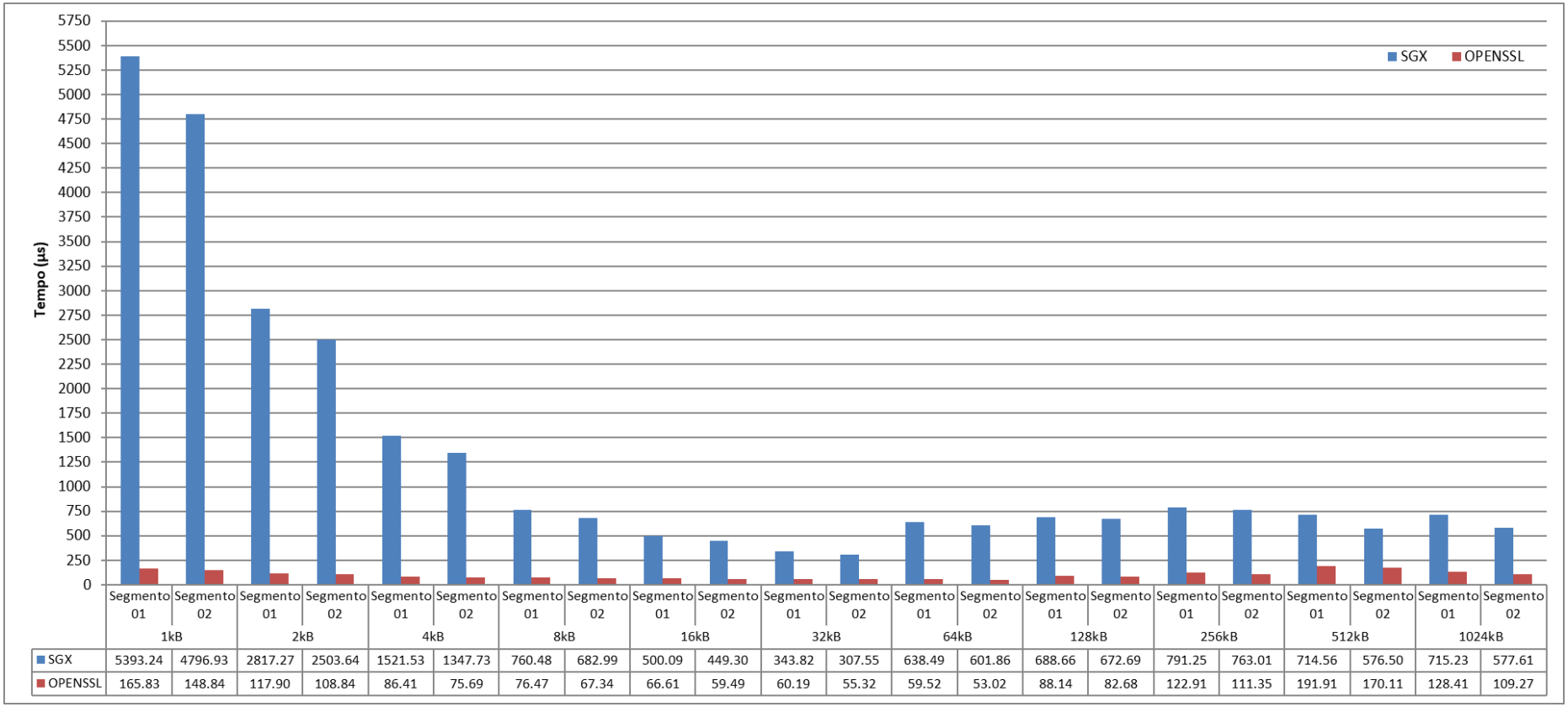
Baseado nesse comportamento atípico, a análise de desempenho de descriptografia deste Estudo de Caso deu-se apenas usando uma janela com o tamanho dos *buffers* de entrada de dados de 16kB, 32kB e 64kB.

A Figura 18 mostra um gráfico em detalhes que, com os tamanhos de *buffers* de entrada de dados de 16kB e 32kB há um decréscimo de processamento na rotina de descriptografia no SGX. E na Figura 19, mostra um gráfico com os tamanhos de *buffers* de entrada de dados de 32kB e 64kB um acréscimo de processamento.

Em relação ao OPENSSL, o tempo de processamento manteve-se estatisticamente igual para os tamanhos de *buffers* de entrada de dados de 16kB, 32kB e 64kB, ficando na média de 50*us*, conforme as Figuras 18 e 19.

Por outro lado, em relação ao tamanho do *buffer* de entrada de dados, os resultados com 32kB tiveram um desempenho melhor que 16kB e 64kB nos dois casos (OPENSSL e SGX).

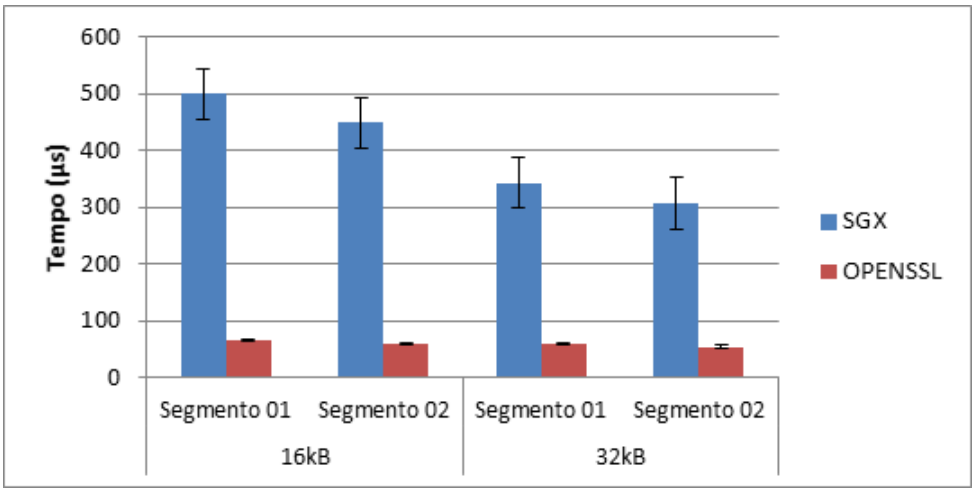
Figura 17 - Duração média de descriptografia em SGX e OPENSSL (Vídeo *"Coast Guard"*) com respectivos tamanhos de *buffers* de entrada



	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02
	1kB		2kB		4kB		8kB		16kB		32kB		64kB		128kB		256kB		512kB		1024kB	
SGX	5393.24	4796.93	2817.27	2503.64	1521.53	1347.73	760.48	682.99	500.09	449.30	343.82	307.55	638.49	601.86	688.66	672.69	791.25	763.01	714.56	576.50	715.23	577.61
OPENSSL	165.83	148.84	117.90	108.84	86.41	75.69	76.47	67.34	66.61	59.49	60.19	55.32	59.52	53.02	88.14	82.68	122.91	111.35	191.91	170.11	128.41	109.27

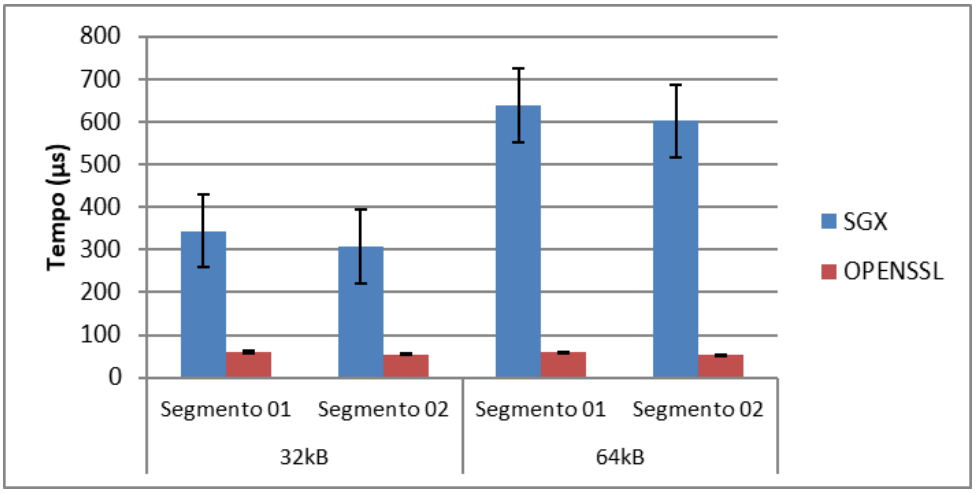
Fonte: Autoria Própria

Figura 18 - Duração média de descriptografia para 16 e 32kB (Vídeo *"Coast Guard"*)



Fonte: Autoria Própria

Figura 19 - Duração média de descriptografia para 32 e 64kB (Vídeo *"Coast Guard"*)



Fonte: Autoria Própria

Pelos resultados obtidos com a diminuição do desempenho de 32kB para 64kB, acredita-se que eles se devem ao fato de que a memória EPC (*Enclave Page Cache*) possui um tamanho de página inferior a 64kB, isto é, de 32kB. E a partir de 64kB, a *enclave* precisa alocar mais páginas de memória para realizar suas operações. Esse mesmo comportamento foi observado nos experimentos de desempenho realizado por (GJERDRUM et al., 2017).

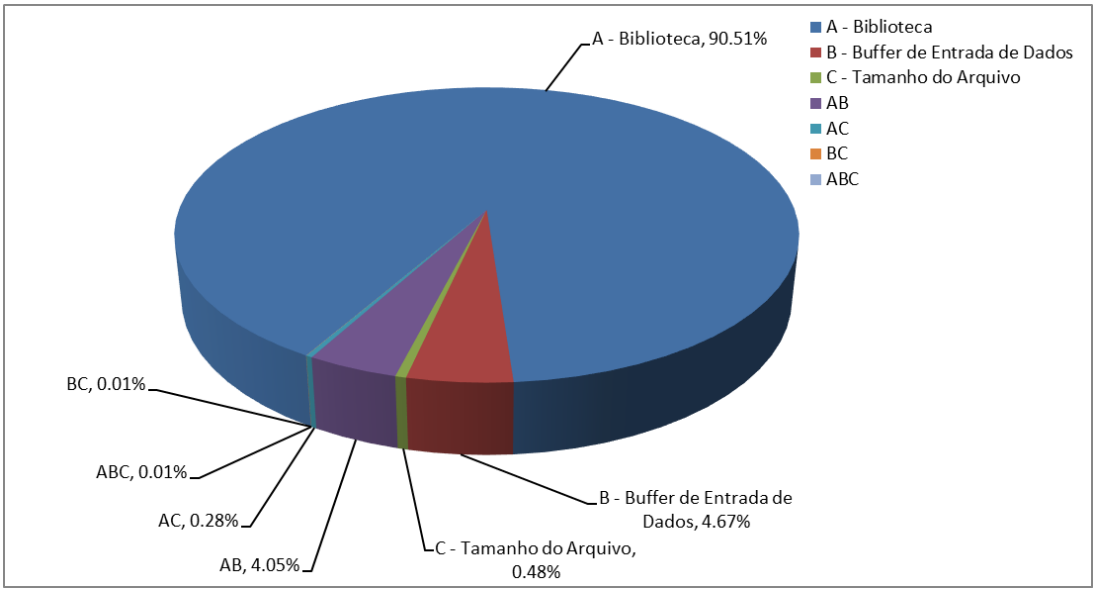
Além disso, (GJERDRUM et al., 2017) aponta várias recomendações como a de utilizar uma tamanho de *buffer* de entrada de dados inferior a 64kB, ou seja, no máximo de 32kB, para um ótimo desempenho com *enclaves*.

Uma análise de influência de três fatores importantes foi realizada. Os resultados obtidos quando se considera tamanho de *buffers* de entrada de dados de 16kB e 32kB são

apresentados na Figura 20, ou seja:

- o tipo de biblioteca (Fator A) influenciou mais os resultados, representando 90,51%;
- o tamanho do *buffer* de entrada (Fator B) influenciou 4,67%;
- a combinação do Fator A+B influenciou 4,05%;
- o tamanho do arquivo (Fator C) e as combinações envolvendo os Fatores A+C, B+C e A+B+C tiveram influências irrelevantes.

Figura 20 - Influência de vários fatores na descriptografia de 16 e 32kB (Vídeo *"Coast Guard"*)



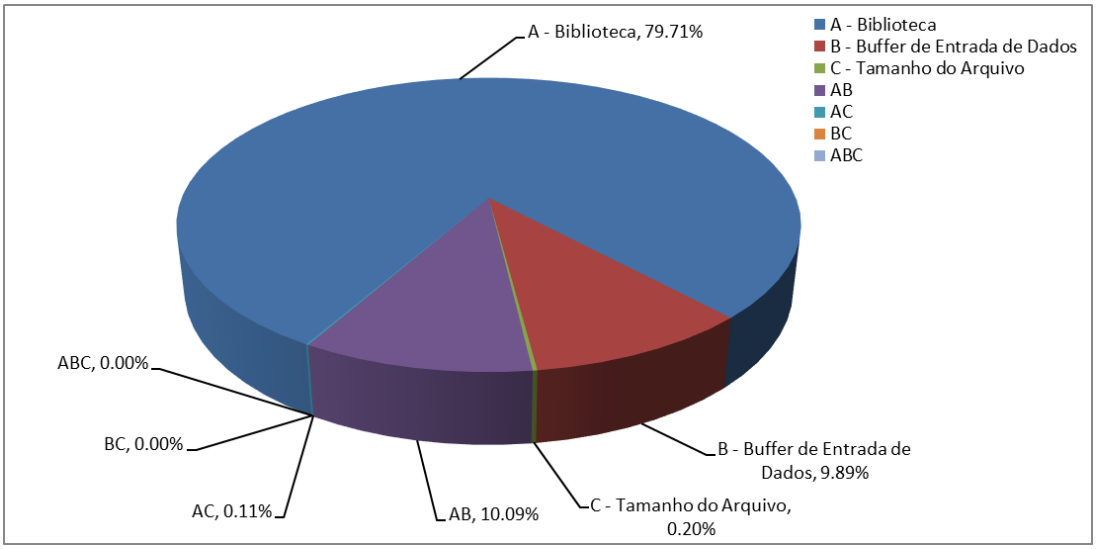
Fonte: Autoria Própria

A Figura 21 mostra os resultados referente ao tamanho de *buffers* de entrada de dados de 32kB e 64kB:

- o tipo de biblioteca (Fator A) influenciou mais os resultados, representando 79,71%;
- o tamanho do *buffer* de entrada (Fator B) influenciou 9,89%;
- a combinação do Fator A+B influenciou 10,09%;
- o tamanho do arquivo (Fator C) e as combinações envolvendo os Fatores A+C, B+C e A+B+C tiveram influências irrelevantes.

Como apontado anteriormente, embora uma técnica de criptografia de *hardware* tenha levado a uma redução de desempenho, ainda é viável para a aplicação final da descriptografia seguida da reprodução de vídeo, melhorando a segurança dos dados transmitidos.

Figura 21 - Influência de vários fatores na descriptografia de 32 e 64kB (Vídeo *"Coast Guard"*)



Fonte: Autoria Própria

Os resultados mostrados na Figura 22 permite a análise de tempos para a execução de cada etapa:

- Os tempos de descriptografia dos segmentos de vídeo 01 e 02 e com o tamanho do *buffer* de entrada de dados de 32kB tiveram o melhor desempenho em relação aos demais;
- Não houve diferença significativa no tempo de descriptografia do SGX para os três *buffers* de entrada de dados;
- O que mais afetou o desempenho foi a entrada na *enclave* e o processamento interno na *enclave*, indicando que provavelmente as operações de ECALL sejam os grandes vilões do desempenho.

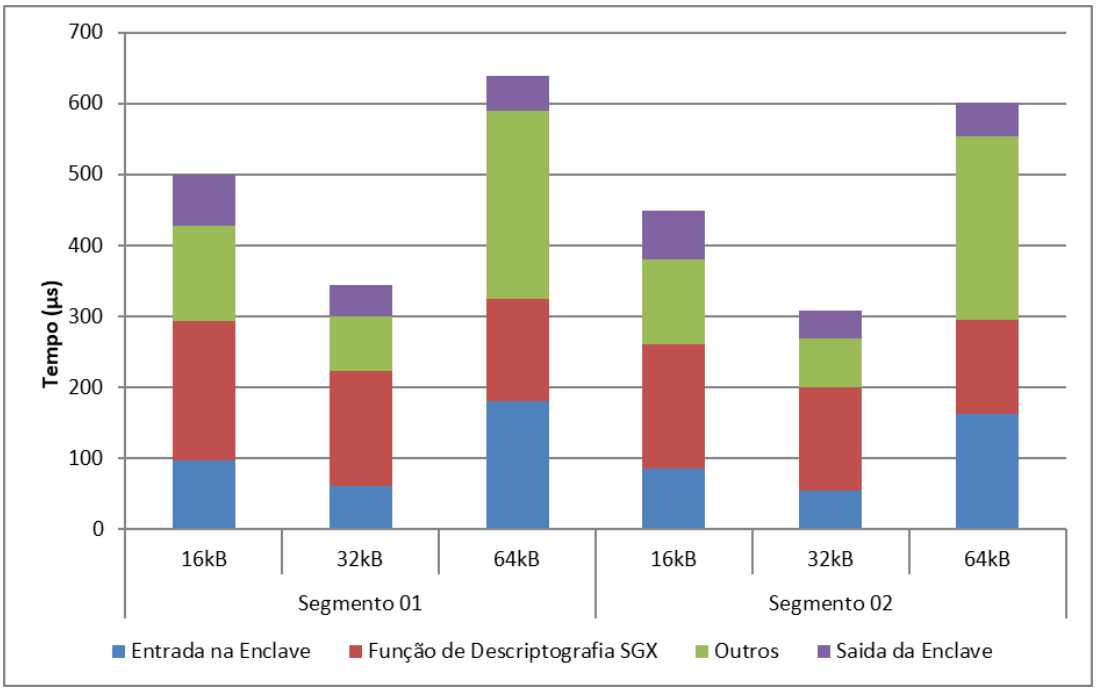
4.1.2 ESTUDO DE CASO 2: VÍDEO DE RESOLUÇÃO 720P (*DUCKS TAKE OFF*)

Neste segundo estudo de caso, um vídeo de resolução de 1280x720 de entrelaçamento progressivo, chamado *Ducks Take Off*, é analisado .

Para iniciar a análise dos resultados deste segundo estudo de caso, a Figura 23 mostra amplamente o desempenho de diferentes tamanhos de *buffer* de entrada de dados, na transferência de dados de/para a função de descriptografia dentro da *enclave*.

Observa-se que de todos os tamanhos de *buffer* de entrada de dados, o de 32kB oferece o melhor desempenho, em relação aos demais, considerando o tempo gasto relacionado à execução de várias transferências de dados em pequenas partes para a *enclave*.

Figura 22 - Tempo execução das etapas de descriptografia (Vídeo *"Coast Guard"*)



Fonte: Autoria Própria

Observa-se na Figura 23, apenas com resultados do SGX, que:

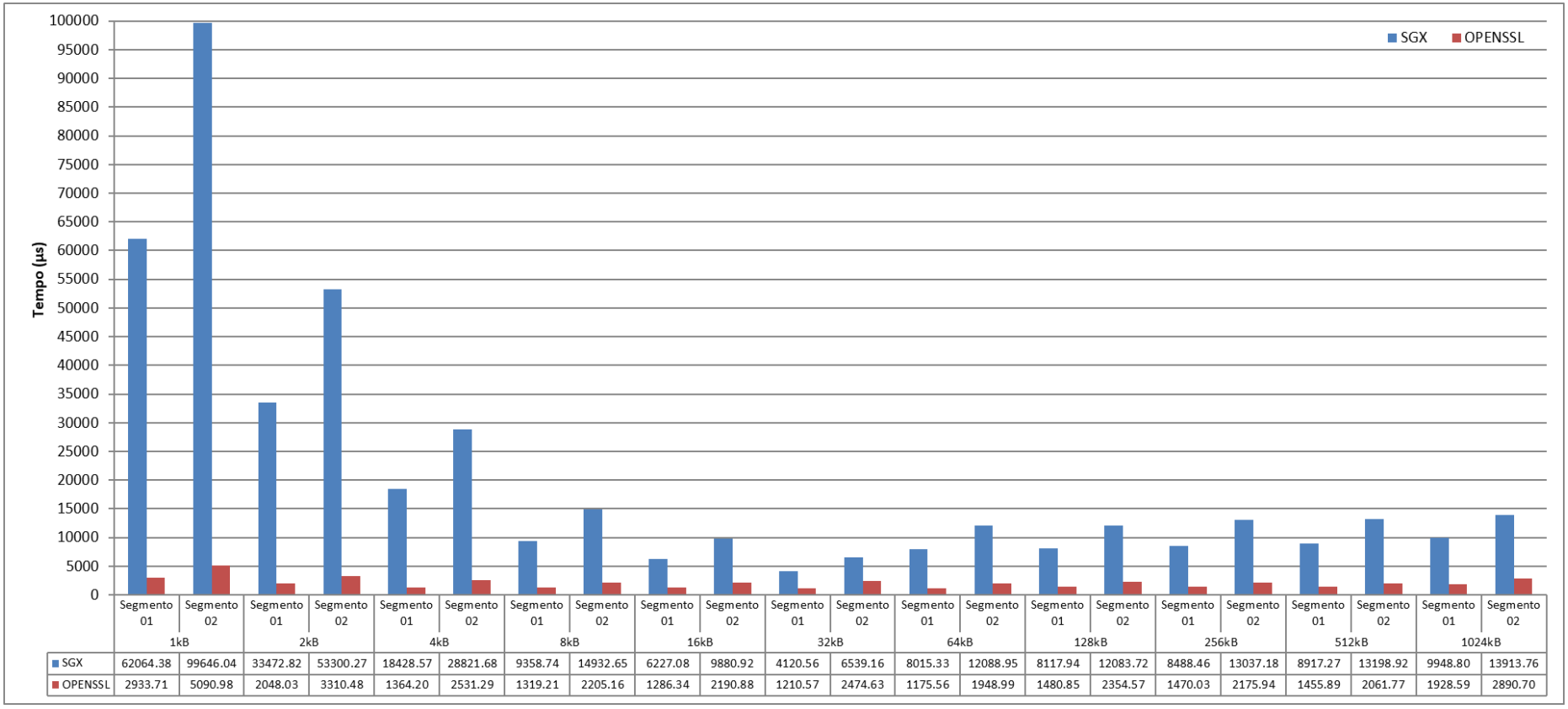
- O desempenho melhora a medida que o tamanho dos *buffers* de entrada de dados aumenta de 1kB à 32kB, já que há uma redução no processo de descriptografia;
- O desempenho piora a medida que o tamanho dos *buffers* de entrada de dados aumenta de 32kB para 64kB. E a partir de 64kB até chegar em 1024kB, há um tempo constante no processo de descriptografia.

O comportamento de degradação do desempenho mostrado na Figura 23 e já citado no Estudo de Caso 1 da subseção 4.1.1, a análise de desempenho de descriptografia deste Estudo de Caso dar-se-á entre 16kB, 32kB e 64kB.

A Figura 24 mostra um gráfico em detalhes que, com os tamanhos de *buffers* de entrada de dados de 16kB e 32kB há um decréscimo de processamento na rotina de descriptografia no SGX. E na Figura 25, mostra um gráfico com os tamanhos de *buffers* de entrada de dados de 32kB e 64kB um acréscimo de processamento.

Em relação ao OPENSSL, o tempo de processamento mantém um diferença de 1*ms* entre os segmentos de vídeo 1 e 2 para os tamanhos de *buffers* de entrada de dados de 16kB, 32kB e 64kB conforme mostram as Figuras 24 e 25.

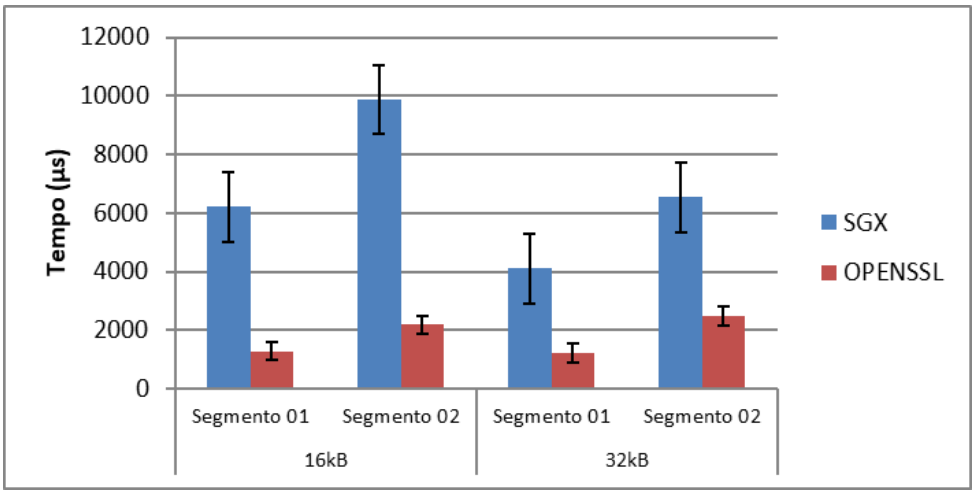
Figura 23 - Duração média de descriptografia em SGX e OPENSSL (Vídeo *"Ducks Take Off"*) com respectivos tamanhos de *buffers* de entrada



	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02
	1kB		2kB		4kB		8kB		16kB		32kB		64kB		128kB		256kB		512kB		1024kB			
SGX	62064.38	99646.04	33472.82	53300.27	18428.57	28821.68	9358.74	14932.65	6227.08	9880.92	4120.56	6539.16	8015.33	12088.95	8117.94	12083.72	8488.46	13037.18	8917.27	13198.92	9948.80	13913.76		
OPENSSL	2933.71	5090.98	2048.03	3310.48	1364.20	2531.29	1319.21	2205.16	1286.34	2190.88	1210.57	2474.63	1175.56	1948.99	1480.85	2354.57	1470.03	2175.94	1455.89	2061.77	1928.59	2890.70		

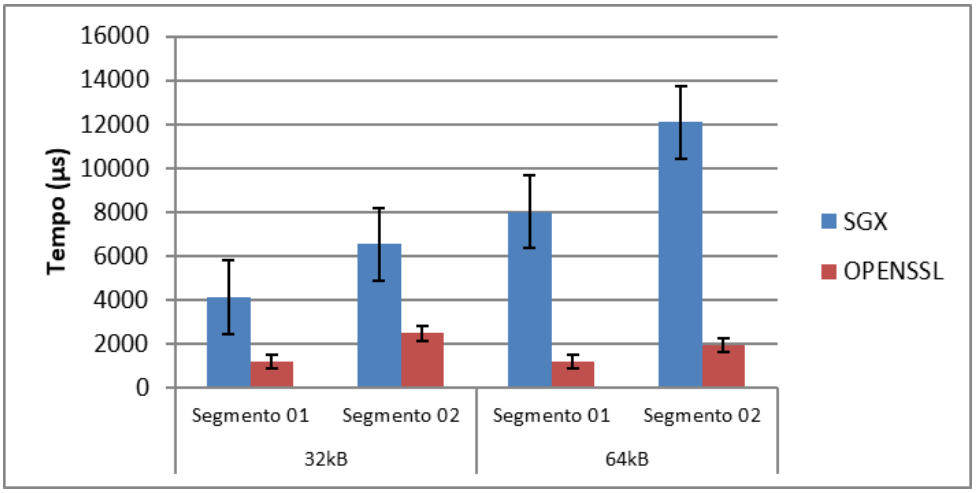
Fonte: Autoria Própria

Figura 24 - Duração média de descriptografia para 16 e 32kB (Vídeo *"Ducks Take Off"*)



Fonte: Autoria Própria

Figura 25 - Duração média de descriptografia para 32 e 64kB (Vídeo *"Ducks Take Off"*)



Fonte: Autoria Própria

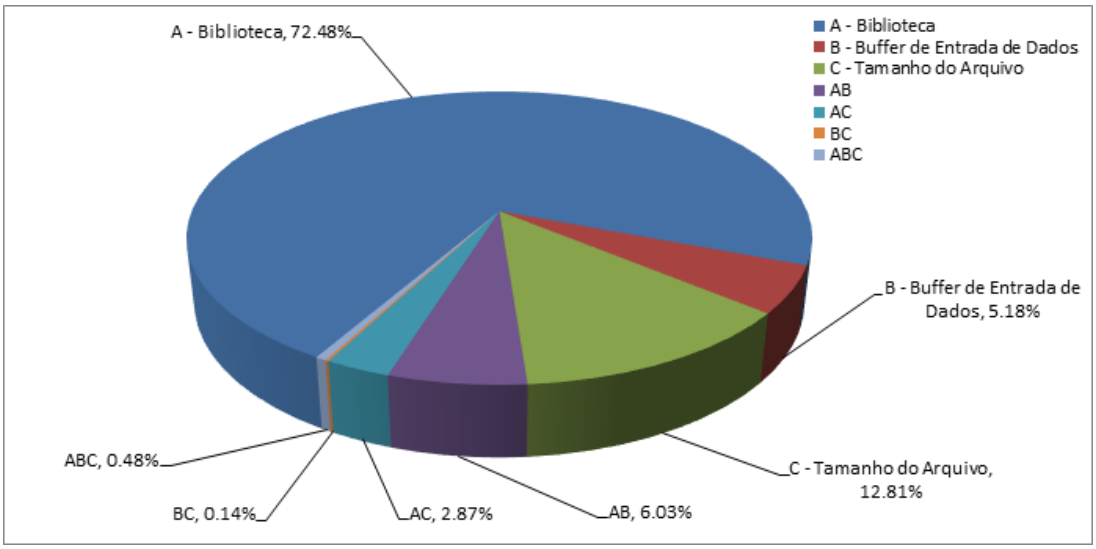
Por outro lado, em relação ao tamanho do *buffer* de entrada de dados, os resultados com 32kB tiveram desempenhos melhores que o de 16kB e 64kB (OPENSSL e SGX).

Através dos resultados obtidos com a diminuição do desempenho de 32kB para 64kB, deve-se ao fato de que a memória EPC (*Enclave Page Cache*) possua um tamanho de página inferior a 64kB. Justificativa já mencionada no Estudo de Caso 1 na subseção 4.1.1 através dos experimentos de desempenho realizado por (GJERDRUM et al., 2017).

Uma análise de influência de três fatores importantes foi realizada. Os resultados obtidos quando se considera tamanho de *buffers* de entrada de dados de 16kB e 32kB são apresentados na Figura 26, ou seja:

- o tipo de biblioteca (Fator A) influenciou mais os resultados, representando 72,48%;
- o tamanho do arquivo (Fator C), com 12,81%, foi o segundo fator que mais influenciou no resultado final. A justificativa é devido que o vídeo seja em alta resolução, isto é, contém mais dados de vídeo se comparado a um vídeo de resolução padrão (Estudo de Caso 1 na subseção 4.1.1);
- a combinação do Fator A+B influenciou 6,03%, com peso maior do Fator A;
- o *buffer* de entrada de dados, Fator B com 5,18%, foi o quarto fator que mais influenciou no resultado final;
- a combinação do Fator A+C influenciou 2,87%, com peso maior do Fator A;
- a combinação dos Fatores B+C e A+B+C influenciaram muito pouco, respectivamente 0,14% e 0,48%.

Figura 26 - Influência de vários fatores na descriptografia de 16 e 32kB (Vídeo *"Ducks Take Off"*)



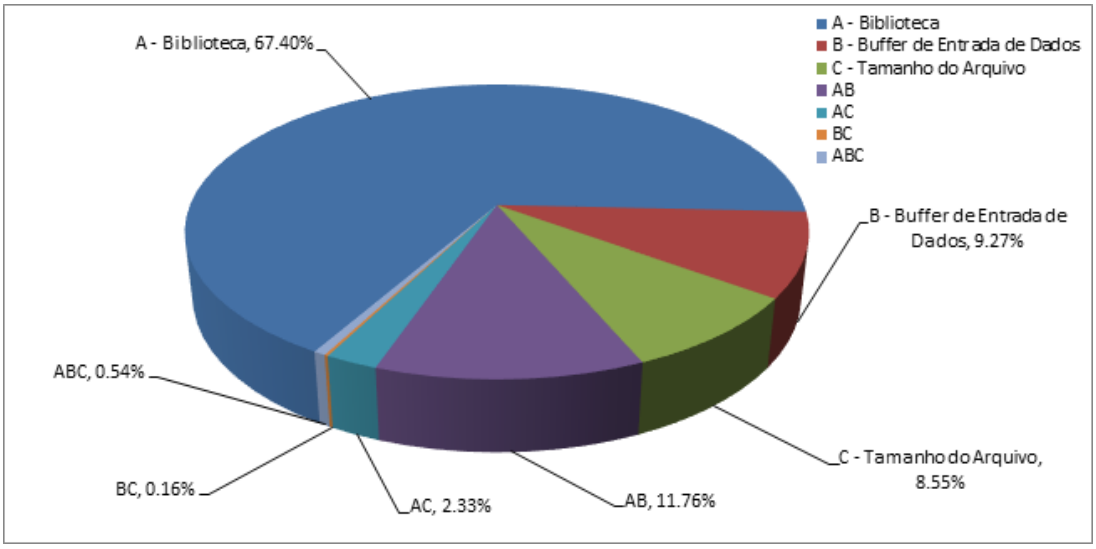
Fonte: Autoria Própria

Os resultados obtidos quando se considera tamanho de *buffers* de entrada de dados de 32kB e 64kB são apresentados na Figura 27, ou seja:

- o tipo de biblioteca (Fator A) influenciou mais os resultados, representando 67,40%;
- a combinação do Fator A+B influenciou 11,76%, com peso maior do Fator A;
- o *buffer* de entrada de dados (Fator B), com 9,27%, foi o terceiro fator que mais influenciou no resultado final;

- o tamanho do arquivo (Fator C), com 8,55%;
- a combinação do Fator A+C influenciou 2,33%, com peso maior do Fator A;
- a combinação dos Fatores B+C e A+B+C tiveram influências irrelevantes, respectivamente 0,16% e 0,54%.

Figura 27 - Influência de vários fatores na descriptografia de 32 e 64kB (Vídeo *"Ducks Take Off"*)

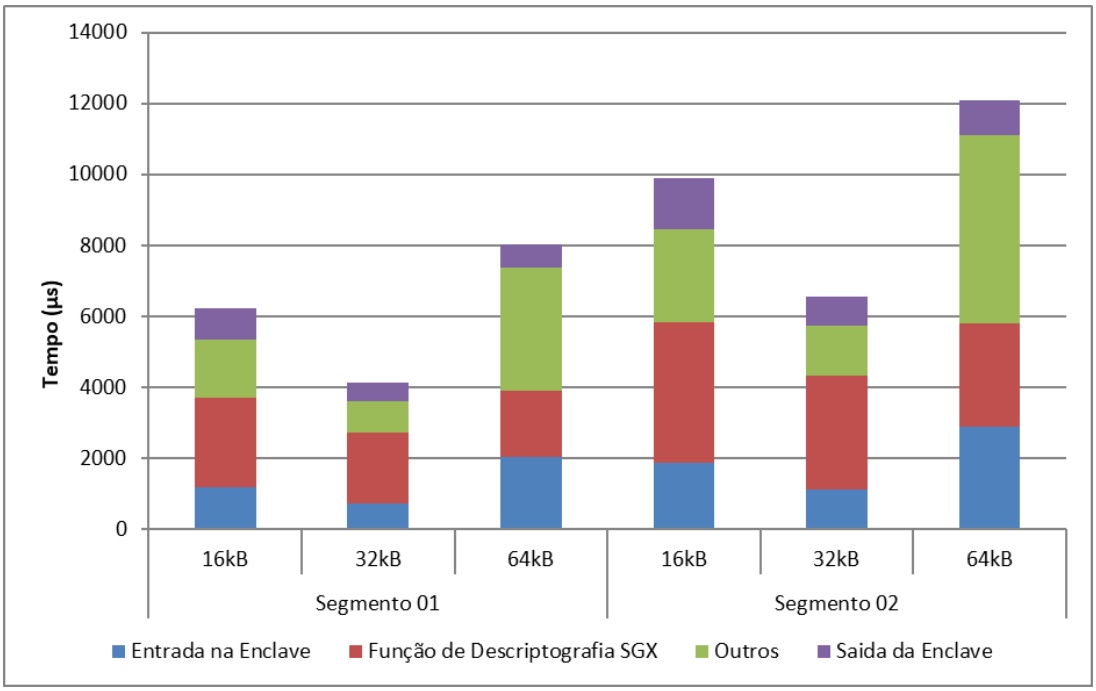


Fonte: Autoria Própria

Os resultados mostrados na Figura 28 permite a análise de tempos para a execução de cada etapa:

- O tempo de descriptografia do segmento de vídeo 01 com o tamanho do *buffer* de entrada de dados de 32kB, teve o melhor desempenho em relação aos demais;
- O tempo de descriptografia do segmento de vídeo 02 com o tamanho do *buffer* de entrada de dados de 16kB, teve o pior desempenho em relação aos demais;
- O tempo de processamento interno (OUTROS) na *enclave* do segmento de vídeo 02 com o tamanho do *buffer* de entrada de dados de 64kB, teve o pior desempenho em relação aos demais;
- Não houve uma diferença muita significativa nas operações de ECALL e OCALL do SGX para os três *buffers* de entrada de dados.

Figura 28 - Tempo execução das etapas de descriptografia (Vídeo *"Ducks Take Off"*)



Fonte: Autoria Própria

4.1.3 ESTUDO DE CASO 3: VÍDEO DE RESOLUÇÃO 1080P (*IN TO TREE*)

Neste terceiro estudo de caso, um vídeo de resolução de 1920x1080 de entrelaçamento progressivo, chamado *In To Tree*, é analisado .

Para iniciar a análise dos resultados deste terceiro estudo de caso, a Figura 29 mostra amplamente o desempenho de diferentes tamanhos de *buffer* de entrada de dados, na transferência de dados de/para a função de descriptografia dentro da *enclave*.

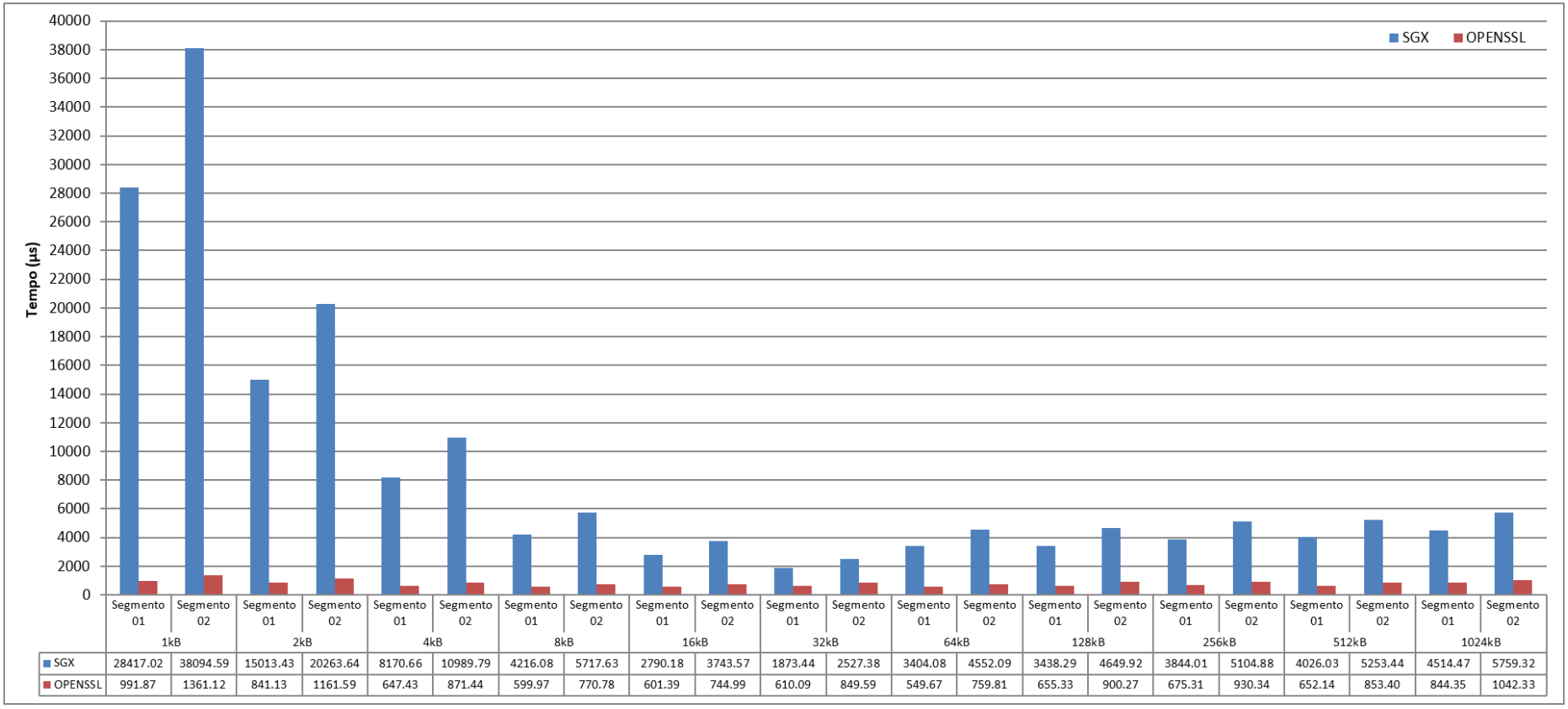
Observa-se que de todos os tamanhos de *buffer* de entrada de dados, o de 32kB oferece o melhor desempenho, em relação aos demais, considerando o tempo gasto relacionado à execução de várias transferências de dados em pequenas partes para a *enclave*.

Observa-se na Figura 29, apenas com resultados do SGX, que:

- O desempenho melhora a medida que o tamanho dos *buffers* de entrada de dados aumenta de 1kB à 32kB, já que há uma redução no processo de descriptografia;
- O desempenho piora a medida que o tamanho dos *buffers* de entrada de dados aumenta de 32kB para 64kB. E a partir de 64kB até chegar em 1024kB, há um tempo constante no processo de descriptografia.

O comportamento de degradação do desempenho mostrado na Figura 29 e já citado

Figura 29 - Duração média de descriptografia em SGX e OPENSSL (Vídeo *"In To Tree"*) com respectivos tamanhos de *buffers* de entrada



	1kB		2kB		4kB		8kB		16kB		32kB		64kB		128kB		256kB		512kB		1024kB	
	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02
SGX	28417.02	38094.59	15013.43	20263.64	8170.66	10989.79	4216.08	5717.63	2790.18	3743.57	1873.44	2527.38	3404.08	4552.09	3438.29	4649.92	3844.01	5104.88	4026.03	5253.44	4514.47	5759.32
OPENSSL	991.87	1361.12	841.13	1161.59	647.43	871.44	599.97	770.78	601.39	744.99	610.09	849.59	549.67	759.81	655.33	900.27	675.31	930.34	652.14	853.40	844.35	1042.33

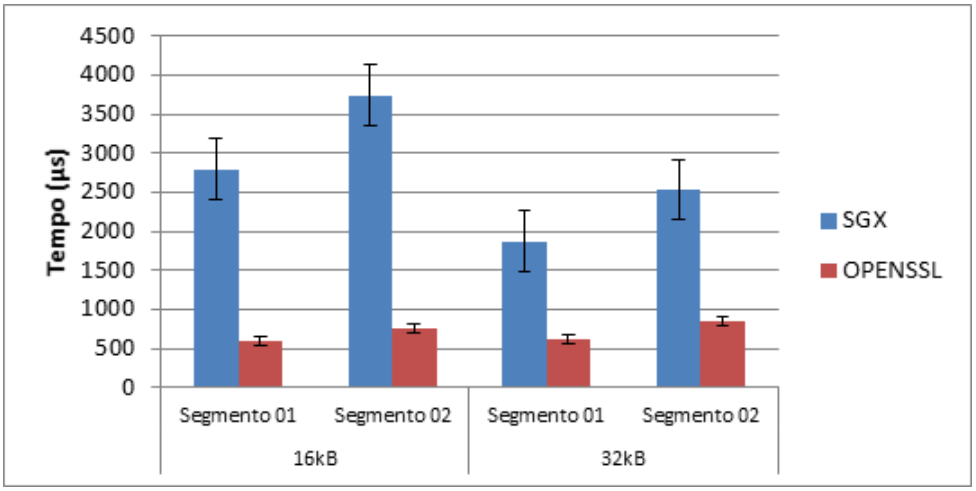
Fonte: Autoria Própria

nos Estudos de Caso 1 e 2, a análise de desempenho de descriptografia deste Estudo de Caso dar-se-á entre 16kB, 32kB e 64kB.

A Figura 30 mostra um gráfico em detalhes que, com os tamanhos de *buffers* de entrada de dados de 16kB e 32kB há um decréscimo de processamento na rotina de descriptografia no SGX. E na Figura 31, mostra um gráfico com os tamanhos de *buffers* de entrada de dados de 32kB e 64kB um acréscimo de processamento.

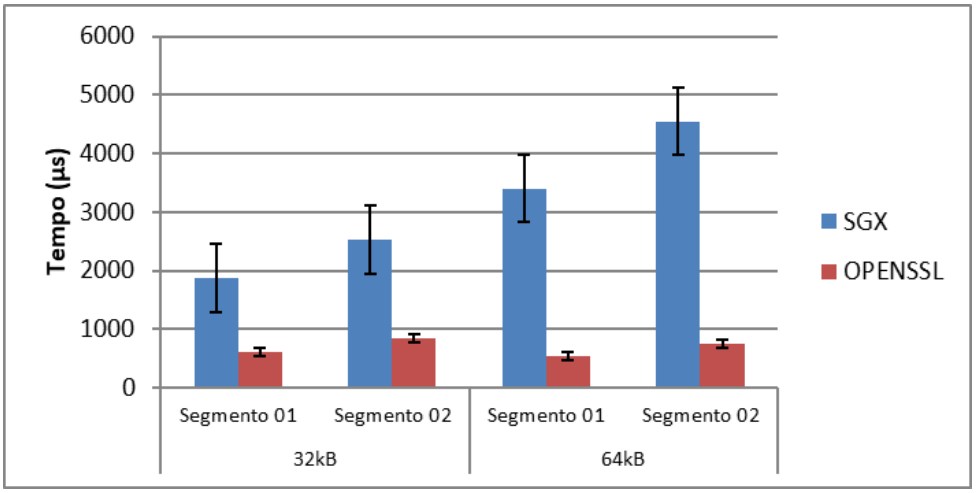
Em relação ao OPENSSL, o tempo de processamento mantém-se constante para os tamanhos de *buffers* de entrada de dados de 16kB, 32kB e 64kB conforme mostram as Figuras 30 e 31.

Figura 30 - Duração média de descriptografia para 16 e 32kB (Vídeo *"In To Tree"*)



Fonte: Autoria Própria

Figura 31 - Duração média de descriptografia para 32 e 64kB (Vídeo *"In To Tree"*)



Fonte: Autoria Própria

Por outro lado, em relação ao tamanho do *buffer* de entrada de dados, os resultados

com 32kB tiveram desempenhos melhores que o de 16kB e 64kB (OPENSSL e SGX).

Através dos resultados obtidos com a diminuição do desempenho de 32kB para 64kB, deve-se ao fato de que a memória EPC (*Enclave Page Cache*) possua um tamanho de página inferior a 64kB. A justificativa é a mesma mencionada nos Estudos de Casos 1 e 2 através dos experimentos de desempenho realizado por (GJERDRUM et al., 2017).

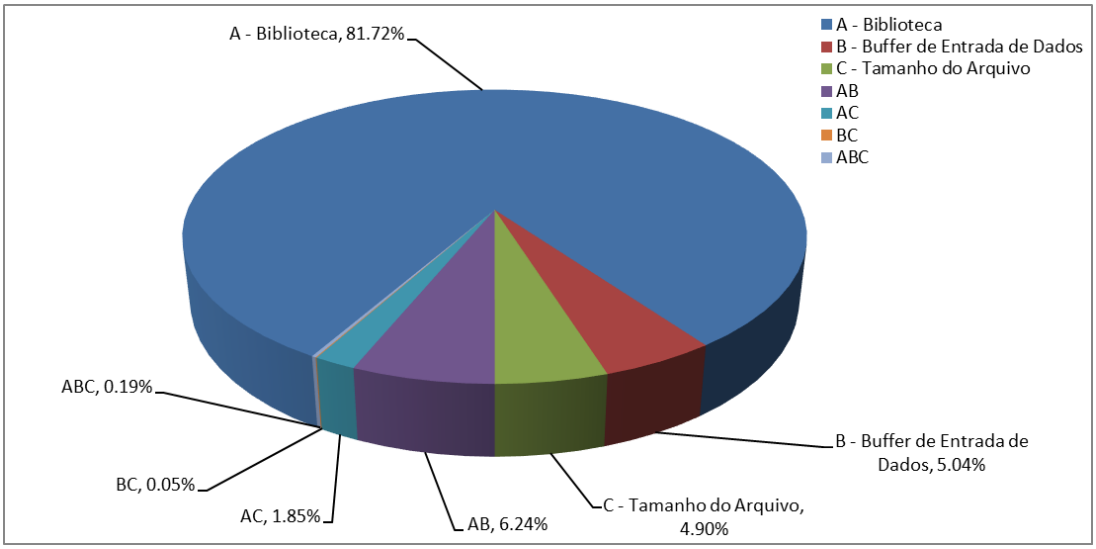
Além disso, uma análise de influência de três fatores importantes foi realizada. Os resultados obtidos quando se considera tamanho de *buffers* de entrada de dados de 16kB e 32kB são apresentados na Figura 32, ou seja:

- o tipo de biblioteca (Fator A) influenciou mais os resultados, representando 81,72%;
- a combinação do Fator A+B influenciou 6,24%, com peso maior do Fator A;
- o tamanho do arquivo (Fator C), com 4,90%, foi o terceiro fator que mais influenciou no resultado final. A justificativa é devido que o vídeo seja em alta resolução, isto é, contém mais dados de vídeo se comparado a um vídeo de resolução padrão (Estudo de Caso 1);
- a combinação do Fator A+B influenciou 6,24%, com peso maior do Fator A;
- o *buffer* de entrada de dados, Fator B, com 5,04%;
- a combinação do Fator A+C influenciou 1,85%, com peso maior do Fator A;
- a combinação dos Fatores B+C e A+B+C influenciaram muito pouco, respectivamente 0,05% e 0,19%.

Os resultados obtidos quando se considera tamanho de *buffers* de entrada de dados de 32kB e 64kB são apresentados na Figura 33, ou seja:

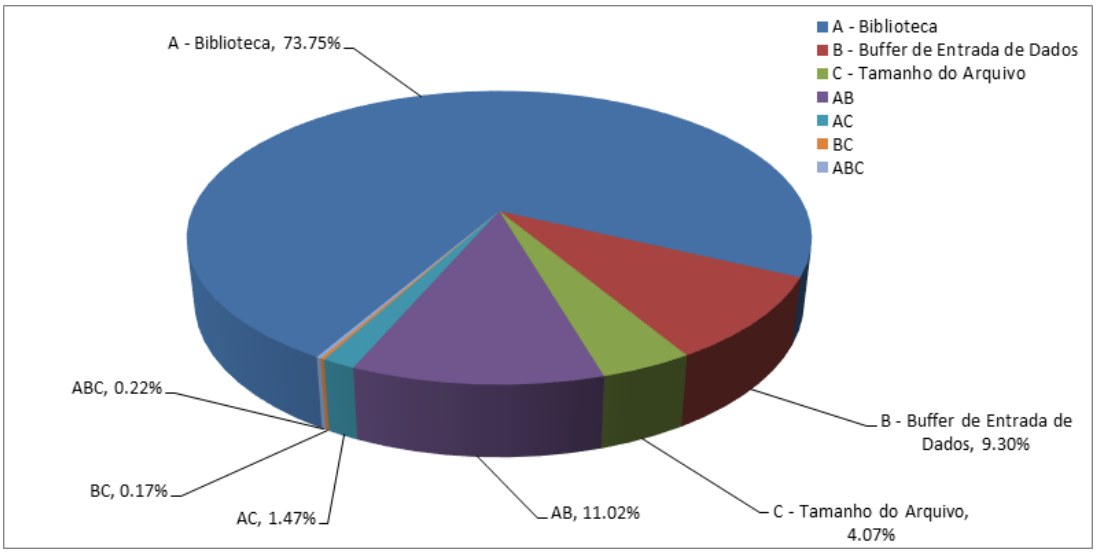
- o tipo de biblioteca (Fator A) influenciou mais os resultados, representando 73,75%;
- a combinação do Fator A+B influenciou 11,02%, com peso maior do Fator A;
- o *buffer* de entrada de dados (Fator B), com 9,30%, foi o terceiro fator que mais influenciou no resultado final;
- o tamanho do arquivo (Fator C), com 4,07%;
- a combinação do Fator A+C influenciou 1,47%, com peso maior do Fator A;
- a combinação dos Fatores B+C e A+B+C tiveram influências irrelevantes, respectivamente 0,17% e 0,22%.

Figura 32 - Influência de vários fatores na descriptografia de 16 e 32kB (Vídeo *"In To Tree"*)



Fonte: Autoria Própria

Figura 33 - Influência de vários fatores na descriptografia de 32 e 64kB (Vídeo *"In To Tree"*)



Fonte: Autoria Própria

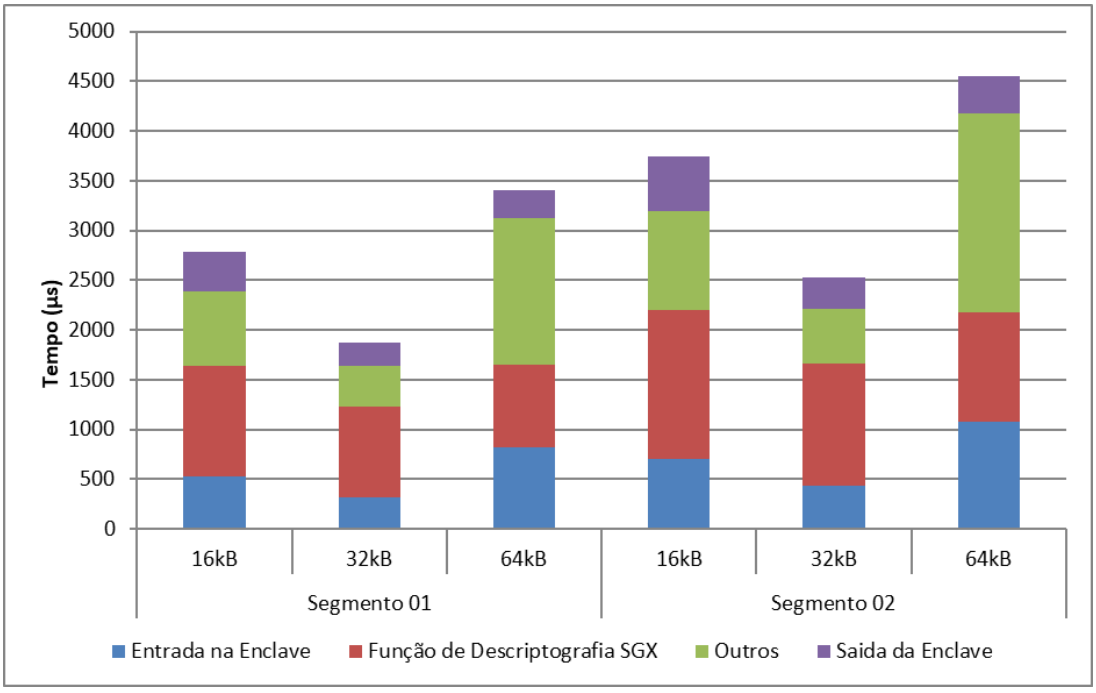
Os resultados mostrados na Figura 34 permite a análise de tempos para a execução de cada etapa:

- O tempo de descriptografia do segmento de vídeo 01 com o tamanho do *buffer* de entrada de dados de 32kB, teve o melhor desempenho em relação aos demais;
- O tempo de descriptografia do segmento de vídeo 02 com o tamanho do *buffer* de entrada de dados de 16kB, teve o pior desempenho em relação aos demais;
- O tempo de processamento interno (OUTROS) na *enclave* do segmento de vídeo 02 com

o tamanho do *buffer* de entrada de dados de 64kB, teve o pior desempenho em relação aos demais;

- Não houve uma diferença muita significativa nas operações de OCALL do SGX para os três *buffers* de entrada de dados.

Figura 34 - Tempo execução das etapas de descriptografia (Vídeo *"In To Tree"*)



Fonte: Autoria Própria

4.1.4 ESTUDO DE CASO 4: VÍDEO DE RESOLUÇÃO 2160P (*OLD TOWN CROSS*)

Neste quarto e último estudo de caso, um vídeo de resolução de 3840x2160 de entrelaçamento progressivo, chamado *Old Town Cross*, é analisado.

Para iniciar a análise dos resultados deste quarto estudo de caso, a Figura 35 mostra amplamente o desempenho de diferentes tamanhos de *buffer* de entrada de dados, na transferência de dados de/para a função de descriptografia dentro da *enclave*.

Observa-se que de todos os tamanhos de *buffer* de entrada de dados, o de 32kB oferece o melhor desempenho, em relação aos demais, considerando o tempo gasto relacionado à execução de várias transferências de dados em pequenas partes para a *enclave*.

Observa-se na Figura 35, apenas com resultados do SGX, que:

- O desempenho melhora a medida que o tamanho dos *buffers* de entrada de dados aumenta de 1kB à 32kB, já que há uma redução no processo de descriptografia;
- O desempenho piora a medida que o tamanho dos *buffers* de entrada de dados aumenta de 32kB para 64kB. E a partir de 64kB até chegar em 1024kB, há um tempo constante no processo de descriptografia.

O comportamento de degradação do desempenho mostrado na Figura 35 e já citado nos Estudos de Casos 1/2/3, a análise de desempenho de descriptografia deste Estudo de Caso dar-se-á entre 16kB, 32kB e 64kB.

A Figura 36 mostra um gráfico em detalhes que, com os tamanhos de *buffers* de entrada de dados de 16kB e 32kB há um decréscimo de processamento na rotina de descriptografia no SGX. E na Figura 37, mostra um gráfico com os tamanhos de *buffers* de entrada de dados de 32kB e 64kB um acréscimo de processamento.

Em relação ao OPENSSL, o tempo de processamento mantém-se constante para os tamanhos de *buffers* de entrada de dados de 16kB, 32kB e 64kB conforme mostram as Figuras 36 e 37.

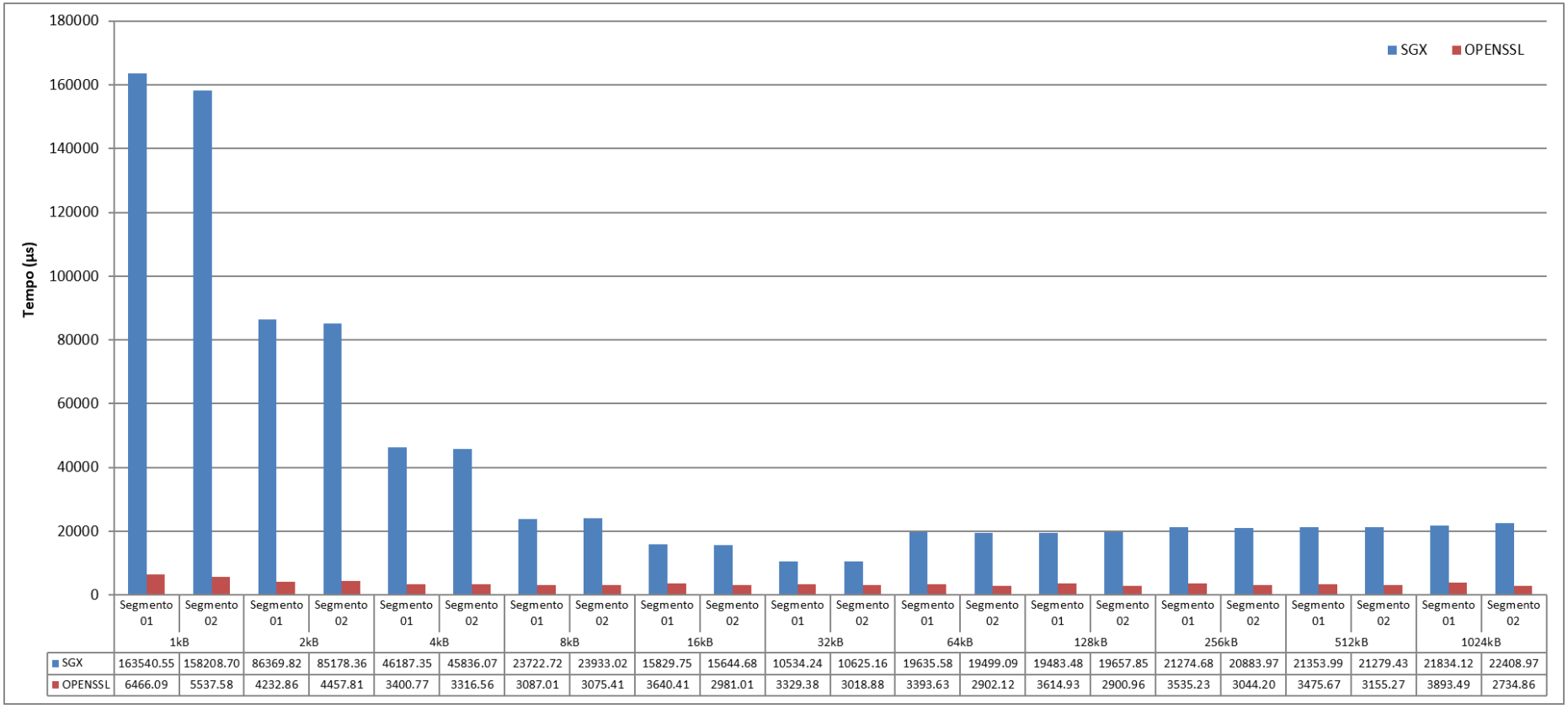
Por outro lado, em relação ao tamanho do *buffer* de entrada de dados, os resultados com 32kB tiveram desempenhos melhores que o de 16kB e 64kB (OPENSSL e SGX).

Através dos resultados obtidos com a diminuição do desempenho de 32kB para 64kB, deve-se ao fato de que a memória EPC (*Enclave Page Cache*) possui um tamanho de página inferior a 64kB, como já observado nos estudos de caso anteriores.

Além disso, uma análise de influência de três fatores importantes foi realizada. Os resultados obtidos quando se considera tamanho de *buffers* de entrada de dados de 16kB e 32kB são apresentados na Figura 38, ou seja:

- o tipo de biblioteca (Fator A) influenciou mais os resultados, representando 87,97%;
- o *buffer* de entrada de dados, Fator B, com 6,27%;
- a combinação do Fator A+B influenciou 5,64%, com peso maior do Fator A;
- o tamanho do arquivo (Fator C) com 0,06%, obteve uma influência irrelevante, mesmo sendo um vídeo de alta resolução, isto é, contendo mais dados de vídeo se comparado a um vídeo de resolução padrão (Estudo de Caso 1);

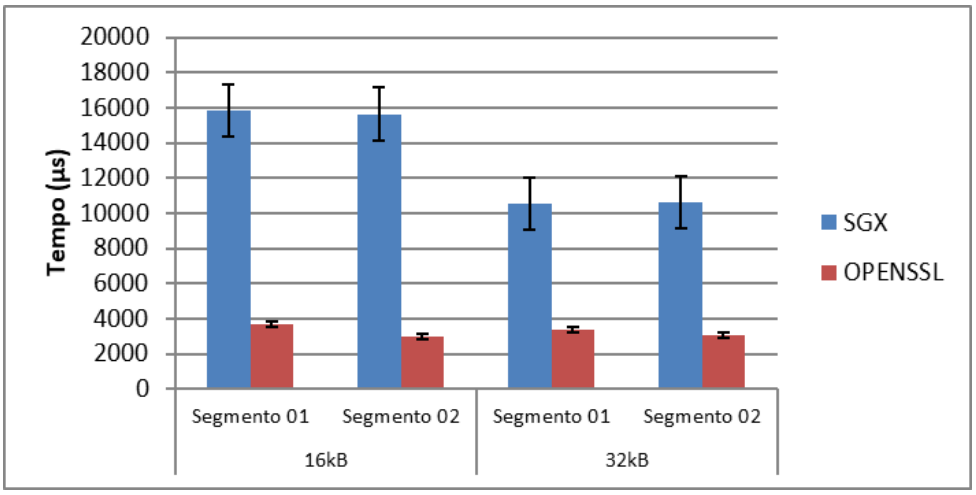
Figura 35 - Duração média de descriptografia em SGX e OPENSSL (Vídeo *"Old Town Cross"*) com respectivos tamanhos de *buffers* de entrada



	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02	Segmento 01	Segmento 02
	1kB		2kB		4kB		8kB		16kB		32kB		64kB		128kB		256kB		512kB		1024kB			
SGX	163540.55	158208.70	86369.82	85178.36	46187.35	45836.07	23722.72	23933.02	15829.75	15644.68	10534.24	10625.16	19635.58	19499.09	19483.48	19657.85	21274.68	20883.97	21353.99	21279.43	21834.12	22408.97		
OPENSSL	6466.09	5537.58	4232.86	4457.81	3400.77	3316.56	3087.01	3075.41	3640.41	2981.01	3329.38	3018.88	3393.63	2902.12	3614.93	2900.96	3535.23	3044.20	3475.67	3155.27	3893.49	2734.86		

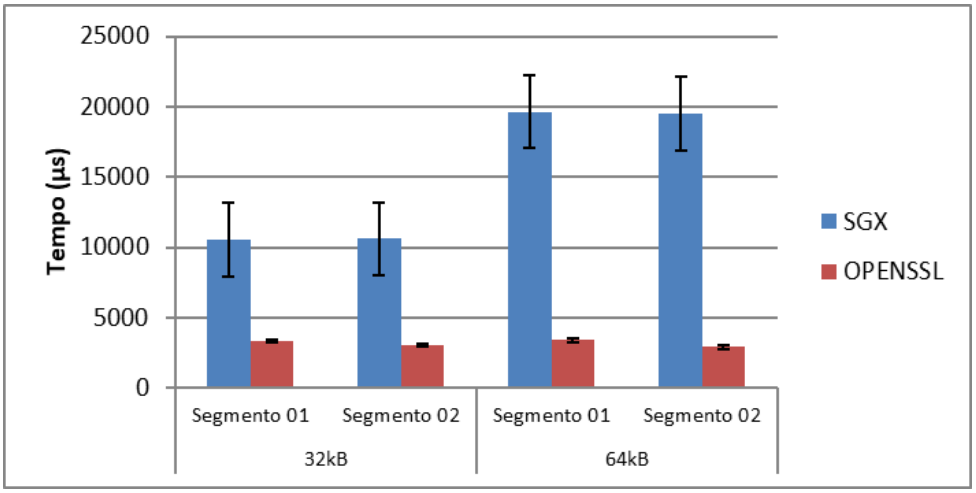
Fonte: Autoria Própria

Figura 36 - Duração média de descriptografia para 16 e 32kB (Vídeo *"Old Town Cross"*)



Fonte: Autoria Própria

Figura 37 - Duração média de descriptografia para 32 e 64kB (Vídeo *"Old Town Cross"*)



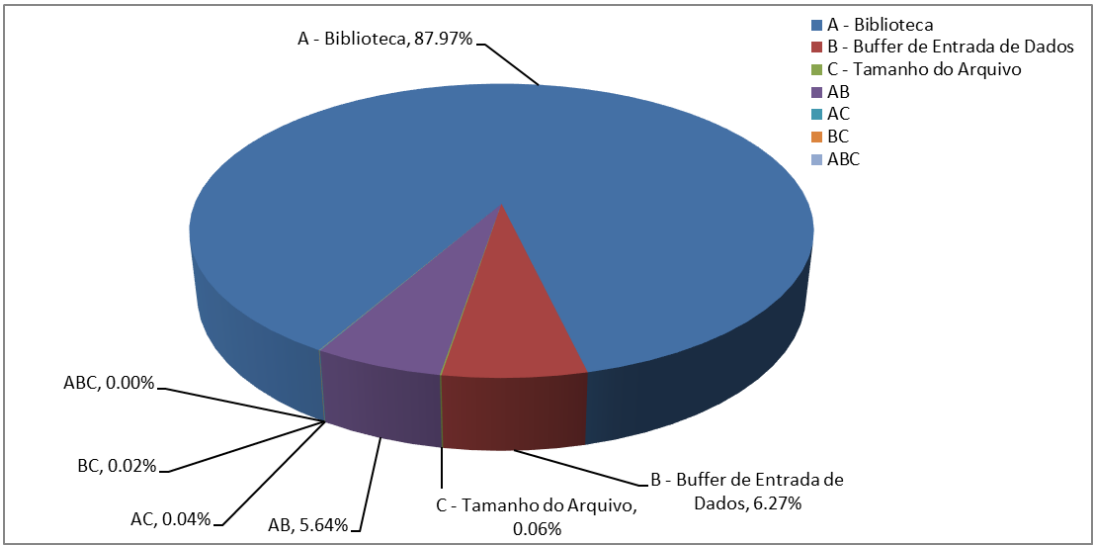
Fonte: Autoria Própria

- a combinação dos Fatores A+C, B+C e A+B+C influenciaram muito pouco, respectivamente 0,04%, 0,02% e 0,00%.

Os resultados obtidos quando se considera tamanho de *buffers* de entrada de dados de 32kB e 64kB são apresentados na Figura 39, ou seja:

- o tipo de biblioteca (Fator A) influenciou mais os resultados, representando 77,81%;
- a combinação do Fator A+B influenciou 11,14%, com peso maior do Fator A;
- o *buffer* de entrada de dados (Fator B) com 11,01%, foi o terceiro fator que mais influenciou no resultado final;

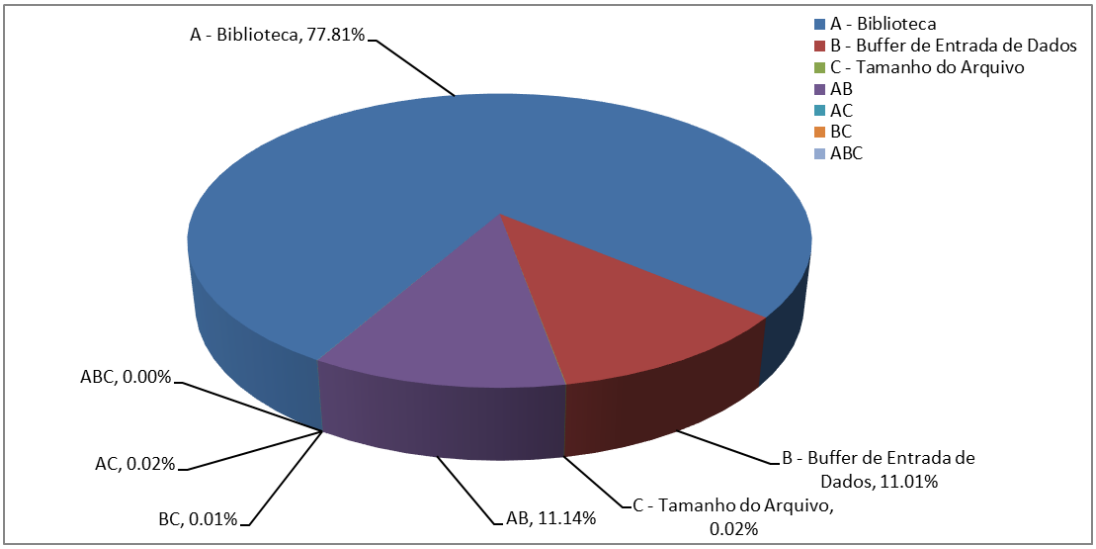
Figura 38 - Influência de vários fatores na descriptografia de 16 e 32kB (Vídeo *"Old Town Cross"*)



Fonte: Autoria Própria

- o tamanho do arquivo (Fator C) com 0,01%, obteve uma influência irrelevante, mesmo sendo um vídeo de alta resolução, isto é, contendo mais dados de vídeo se comparado a um vídeo de resolução padrão (Estudo de Caso 1);
- a combinação dos Fatores A+C, B+C e A+B+C influenciaram muito pouco, respectivamente 0,02%, 0,01% e 0,00%.

Figura 39 - Influência de vários fatores na descriptografia de 32 e 64kB (Vídeo *"Old Town Cross"*)

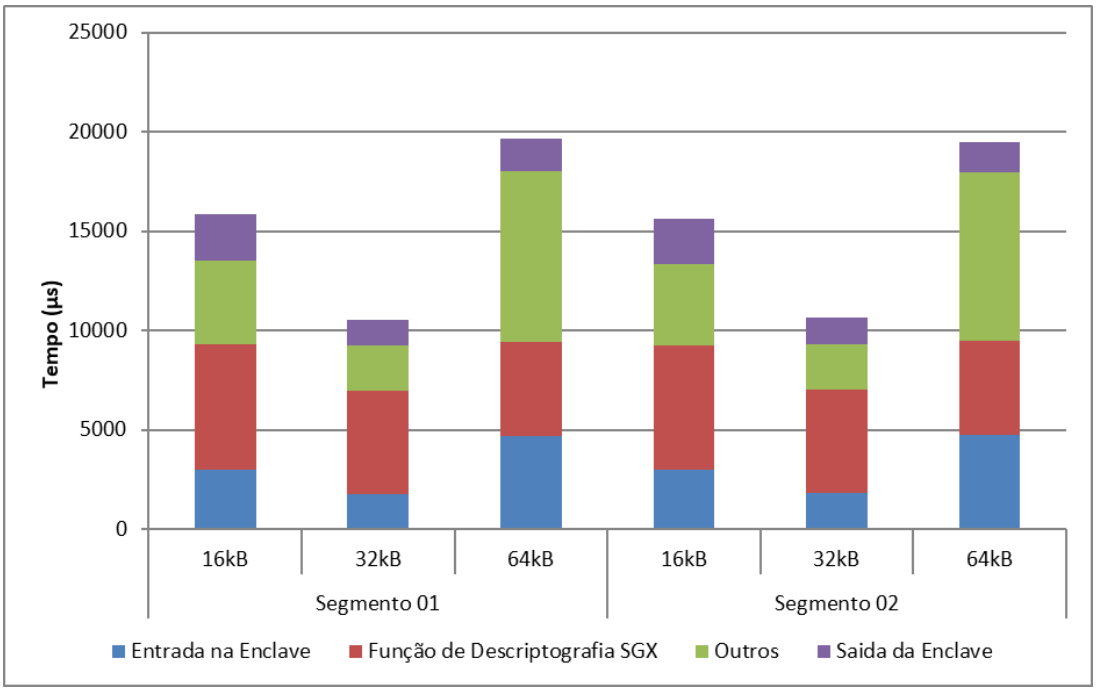


Fonte: Autoria Própria

Os resultados mostrados na Figura 40 permitem a análise de tempos para a execução de cada etapa:

- O tempo de descriptografia total (ECALL, OCALL, Descriptografia SGX e outros processamentos internos da *enclave*), segmento de vídeo 01 com o tamanho do *buffer* de entrada de dados de 32kB, teve o melhor desempenho em relação aos demais;
- O tempo de descriptografia do segmento de vídeo 01 com o tamanho do *buffer* de entrada de dados de 16kB, teve o pior desempenho em relação aos demais;
- O tempo de processamento interno (OUTROS) na *enclave* do segmento de vídeo 02 com o tamanho do *buffer* de entrada de dados de 64kB, teve o pior desempenho em relação aos demais;
- Não houve uma diferença muita significativa nas operações de OCALL do SGX para os três *buffers* de entrada de dados;
- Em relação ao tempo de ECALL, o segmento de vídeo 02 com o tamanho do *buffer* de entrada de dados de 64kB, teve o pior desempenho em relação aos demais.

Figura 40 - Tempo execução das etapas de descriptografia (Vídeo *"Old Town Cross"*)



Fonte: Autoria Própria

4.2 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo foi descrito os resultados obtidos com a implementação de um algoritmo para coletar informações de desempenho de descriptografia de vídeo dentro de uma *enclave*

SGX e também de uma biblioteca chamada OPENSSL, utilizando resoluções distintas de vídeos (480p, 720p, 1080p e 2160p), e a comparação entre o algoritmo de criptografia AES-GCM implementado tanto no SDK do Intel SGX quanto na biblioteca OPENSSL e em diversos tamanhos de *buffers* de entrada (1kB, 2kB, 4kB, 8kB, 16kB, 32kB, 64kB, 128kB, 256kB, 512kB e 1024kB).

E conforme a análise de todos os Estudos de Caso, verificou-se que:

1. Pelos resultados obtidos com a diminuição do desempenho de 32kB para 64kB, comportamento comum a todos os Estudos de Caso, devem-se ao fato de que a memória EPC (*Enclave Page Cache*) possua um tamanho de página inferior a 64kB, isto é, de 32kB. E que a partir de 64kB, a *enclave* precisa alocar mais páginas de memória para realizar suas operações. Esse comportamento foi observado também nos experimentos de desempenho de (GJERDRUM et al., 2017);

2. A análise de influência de três fatores: Tipo de biblioteca (Fator A), tamanho de *buffer* de Entrada de Dados (Fator B) e o tamanho do arquivo (Fator C), Fator C do Estudo de Caso 4 (*Old Town Cross*), obteve uma influência irrelevante, mesmo sendo um vídeo de alta resolução, isto é, contendo mais dados de vídeo se comparado a um vídeo de resolução padrão (Estudo de Caso 1).

O capítulo seguinte apresenta os comentários finais e sugestões para trabalhos futuros.

5 CONCLUSÕES E TRABALHOS FUTUROS

Esta dissertação propôs uma solução simples de proteção de vídeo digital em DRM e distribuído em serviços de VOD utilizando TPM baseado na tecnologia Intel SGX. As simulações foram realizadas em um cenário real e executado no modo seguro, no qual chamamos de *enclave*.

Ressaltando as principais contribuições desta dissertação, podemos citar:

- Desenvolvimento de um sistema de medição de desempenho de descriptografia para o Intel SGX abordando a entrada e saída de dados em uma região de memória segura;
- Implementação de uma solução de criptografia e descriptografia para vídeos remultiplexados no padrão HLS, simulando um ambiente real de cliente de VOD utilizando a arquitetura de *software* do Intel SGX;
- Análise da viabilidade de integração de um TPM em um *player* cliente para o consumo de conteúdos de VOD, especificamente para *Set-Top Boxes*.

Para a avaliação de desempenho do processo de descriptografia de vídeo foram selecionados quatro vídeos, um de cada resolução (480p, 720p, 1080p e 2160p) para mitigar algumas características específicas de cada vídeo.

Foram analisados os resultados da comparação entre o algoritmo de criptografia AES-GCM implementado tanto no SDK do Intel SGX quanto na biblioteca OPENSSL (sem usar *enclaves*) na operação de descriptografia. O tamanho dos *buffers* de entrada (1kB, 2kB, 4kB, 8kB, 16kB, 32kB, 64kB, 128kB, 256kB, 512kB e 1024kB) foram usados para enviar e receber dados da *enclave*.

Nos cálculos estatísticos de desempenho, visou-se obter a máxima informação com o número mínimo experimentos, utilizando o fatorial completo. Os resultados mostrados são a média das quantidades medidas (100 repetições/experimento) e o intervalo de confiança com um nível de confiança de 95%, conforme é definido por (JAIN, 1991).

Os três fatores importantes usados na análise são:

1. **Fator A**: consiste na biblioteca SGX e OPENSSL;

2. **Fator B**: consiste no tamanho dos *buffers* de entrada de dados;

3. **Fator C**: representa o tamanho dos arquivos dos segmentos de vídeo.

O processo completo envolveu etapas de movimentação de dados de uma área não confiável para a *enclave*, descriptografia desses dados e sua movimentação reversa para área não confiável (para projeção efetiva do segmento de vídeo). Visando uma avaliação detalhada do impacto de cada etapa no processo geral, a medição dos tempos foi realizada de acordo com a Figura 16:

- **Etapa 1**: representa o tempo gasto para entrar na área segura (*enclave*);
- **Etapa 2**: é identificada como "Outros" devido ao fato de representar as operações executadas dentro da *enclave* EXCETO a operação de descriptografia;
- **Etapa 3**: é o próprio tempo de descriptografia, isto é, utilizando a função de descriptografia em questão;
- **Etapa 4**: é o tempo gasto para enviar dados de volta para a área não confiável.

Para atender os fatores no cálculo da medição de desempenho, dois segmentos de vídeo foram considerados:

- **Primeiro segmento de vídeo** - contém os parâmetros de vídeo e informações dos segmentos de vídeo subsequentes;
- **Segundo segmento de vídeo** - contém dados de vídeo.

E conforme a análise de todos os Estudos de Caso, mostrou-se que:

1. Pelos resultados obtidos com a diminuição do desempenho de 32kB para 64kB, comportamento comum a todos os Estudos de Caso, devem-se ao fato de que a memória *Enclave Page Cache* (EPC) possua um tamanho de página inferior a 64kB, isto é, de 32kB. E que a partir de 64kB, a *enclave* precisa alocar mais páginas de memória para realizar suas operações. Esse comportamento foi observado também nos experimentos de desempenho de (GJERDRUM et al., 2017);

2. A análise de influência de três fatores: Tipo de biblioteca (Fator A), tamanho de *buffer* de Entrada de Dados (Fator B) e o tamanho do arquivo (Fator C), Fator C do vídeo de *ultra* ultra resolução 2160p (Estudo de Caso 4), obteve uma influência irrelevante, mesmo

sendo um vídeo de alta resolução, isto é, contendo mais dados de vídeo se comparado a um vídeo de resolução padrão (Estudo de Caso 1);

3. Observou-se o impacto no desempenho da descriptografia de vídeo causado pelo processo de transferência de dados de áreas não confiáveis para *enclaves* e vice-versa (ECALL/OCALL) juntamente com a codificação de memória da *enclave*.

4. Observou-se um mínimo de $325\mu s$ ao descriptografar vídeo de resolução padrão 480p (Estudo de Caso 1) utilizando o algoritmo de criptografia padrão AES-GCM em SGX e *buffer* de entrada de dados de 32kB;

5. Observou-se um mínimo de $10ms$ ao descriptografar vídeo de *ultra* alta resolução 2160p (Estudo de Caso 4) utilizando o algoritmo de criptografia padrão AES-GCM em SGX e *buffer* de entrada de dados de 32kB;

Conclui-se que, apesar dos resultados de desempenho de descriptografia no SGX foram inferiores em comparação ao da biblioteca OPENSSL, o SGX é viável pois agrega mais segurança ao vídeo devido o processo de descriptografia ocorrer em um *hardware* seguro.

Além do mais, o tempo de descriptografia baseado no vídeo de *ultra* alta resolução (2160p), que foi de $10ms$, está dentro do limite (< 4 segundos) que um STB deve descriptografar um segmento de vídeo sem comprometer a qualidade na recepção (LEDERER, 2015).

Para trabalhos futuros, podem-se desenvolver soluções usando o Intel SGX para:

1. Lado do Servidor: aplicativos seguros baseados em nuvem nas redes de entrega de conteúdo para criptografar dinamicamente o conteúdo de vídeo ou criar soluções de *software* a serem disponibilizados como serviço, com o intuito de atender outras empresas que não possam investir um alto preço em infraestrutura de criptografia para *streaming* de vídeo;

2. Lado do cliente: proteger os *players* de vídeo utilizados em diversos tipos de dispositivos embarcados, tais como: *computer sticks* e *set-top boxes*, e além de aplicações OTT (*Over The Top*) para: *tablets*, *smartphones* e *SmartTVs* (MENA, 2017).

Outra proposta de trabalho futuro poderia-se criar *plug-ins* seguros para *web browsers*, ou até mesmo fazer parte da *engine* do W3C WebCryptoAPI (W3C, 2017).

Embora os testes tenham sido realizados em PCs, a Intel já disponibilizou dispositivos embarcados com suporte para SGX (a data de lançamento ocorreu após a realização dos

experimentos apresentados neste trabalho). Esses dispositivos ainda são caros em comparação com produtos similares não SGX, mas à medida que se tornam mais acessíveis, novos aplicativos para sistemas embarcados podem ser economicamente viáveis em um futuro próximo.

REFERÊNCIAS

AKHYAR, F.; NASUTION, S. M.; PURBOYO, T. W. Rabbit algorithm for video on demand. **IEEE Asia Pacific Conference on Wireless and Mobile**, p. 208–213, Agosto 2015.

ANATI, I. et al. Innovative technology for cpu based attestation and sealing. **Proceedings of the 2nd international workshop on hardware and architectural support for security and privacy**, Fevereiro 2013.

APPLE. **About HTTP Live Streaming**. 2016. Disponível em: <https://developer.apple.com/library/content/referencelibrary/GettingStarted/AboutHTTPLiveStreaming/about/about.html>.

ARM. **ARM Trust Zone: A system-wide approach to security**. 2016. Disponível em: <https://www.arm.com/products/security-on-arm/trustzone>.

ARRAG, S. et al. Design and implementation a different architectures of mixcolumn in fpga. Dezembro 2014.

ASI. **Conditional Access Systems (CAS)**. 2017. Disponível em: <https://asi.mk/conditional-access-systems-cas/>.

BABCOCK, A.; DEMERS, C. **What is Aspect Ratio?** 2019. Disponível em: <https://www.rtings.com/tv/learn/what-is-the-aspect-ratio-4-3-16-9-21-9>.

BUTIN, D.; WäLDE, J.; BUCHMANN, J. Post-quantum authentication in openssl with hash-based signatures. **Tenth International Conference on Mobile Computing and Ubiquitous Network (ICMU)**, Outubro 2017.

CANONICAL. **Ubuntu**. 2019. Disponível em: <https://ubuntu.com/>.

CHAKRABORTY, P.; DEV, S.; NAGANUR, R. H. Dynamic http live streaming method for live feeds. **IEEE International Conference on Computational Intelligence and Communication Networks**, p. 1394–1398, Fevereiro 2015.

CISCO. **White Paper: The Zettabyte Era: Trends and Analysis**. 2017. Disponível em: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html>.

COSTA, R. de S. et al. Securing video on demand content with sgx: A decryption performance evaluation in client-side. In: **SIMPÓSIO BRASILEIRO EM SEGURANÇA DA INFORMAÇÃO E DE SISTEMAS COMPUTACIONAIS (SBSEG)**. Rio Grande do Norte: SBC, 2018. p. 127–140.

DIEHL, E. **Securing Digital Video - Techniques for DRM and Content Protection**. 1st. ed. Berlim: Springer-Verlag Berlin Heidelberg, 2012. 264 p.

ENCODING.COM. **Digital Rights Management - An Overview**. 2016. Disponível em: <https://www.encoding.com/digital-rights-management-drm/>.

EPIPHAN. **Frame rate and refresh rate: similar but different**. 2019. Disponível em: <https://www.epiphan.com/blog/frame-rate-refresh-rate/>.

FCC. Fcc16-18 notice of proposed rulemaking and memorandum opinion and order. Federal Communications Commission, 2016.

FFMPEG. **Tizen Studio**. 2018. Disponível em: <https://www.ffmpeg.org>.

GJERDRUM, A. T. et al. Performance of trusted computing in cloud infrastructures with intel sgx. In: **Proc. of the 7th Intern. Conf. on Cloud Computing and Services Science.** Porto, Portugal: SCITEPRESS, 2017. p. 696–703.

GNU. **GCC - The GNU Compiler Collection**. 2019. Disponível em: <https://gcc.gnu.org>.

HARNIK, D. **Impressions of Intel SGX performance**. 2017. Disponível em: <https://medium.com/ @danny_harnik/impressions-of-intel-sgx-performance-22442093595a>.

HUANG, S. **H264 profiles**. 2008. Disponível em: <http://blog.mediacoderhq.com/h264-profiles-and-levels/>.

INTEL. **Overview of an intel software guard extensions enclave life cycle.** 2016. Disponível em: <https://software.intel.com/en-us/blogs/2016/12/20/overview-of-an-intel-software-guard-extensions-enclave-life-cycle>.

INTEL. **Intel Software Guard Extensions Evaluation SDK for Linux OS**. 2017. Disponível em: <https://01.org/intel-softwareguard-extensions>.

INTEL. **Trusted Platform Module 2.0 AXXTPMENC8**. 2017. Disponível em: <https://ark.intel.com/content/www/br/pt/ark/products/124461/trusted-platform-module-2-0-axxtpmenc8.html>.

IRAWAN, I. **Video On Demand - Application of Technology**. 2013. Disponível em: <http://www.almuhibbin.com/2013/01/penerapan-teknologi-video-on-demand.html>.

JAIN, R. **The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling**. 1st. ed. USA: Wiley professional computing, Wiley, 1991. 720 p.

JOHNSON, S.; ZIMMERMAN, D.; DEREK, B. **Intel SGX: Debug, Production, Pre-release what's the difference?** 2016. Disponível em: <https://software.intel.com/en-us/blogs/2016/01/07/intel-sgx-debug-production-prelease-whats-the-difference>.

KAMIENSKI, C. A.; SADOK, D.; CAVALCANTE, K. K. D. A. **Simulando a Internet: Aplicações na pesquisa e no ensino.** 2002. Disponível em: <http://www.cin.ufpe.br/ cak/publications/jai2002-capitulo>.

KNIGHT, S. **Intel to enable SGX technology on future Skylake CPUs**. 2015. Disponível em: <http://www.techspot.com/news/62324-intel-enable-sgx-technology-future-skylake-cpus.html>.

KO, M.; KOO, I. **An overview of Interactive Video On Demand Systems**. Dezembro 1996. Disponível em: <http://www.ece.ubc.ca/ irenek/techpaps/vod/vod.html>.

LEDERER, S. **Optimal Adaptive Streaming Formats MPEG-DASH and HLS Segment Length**. 2015. Disponível em: <https://bitmovin.com/mpeg-dash-hls-segment-length/>.

LIEBEHERR, J. Multimedia networks: Issues and challenges. **Computer Magazine**, v. 28, Abril 1995.

MENA, I. **Verbete Draft: o que é OTT**. 2017. Disponível em: <https://projetodraft.com/verbete-draft-o-que-e-ott/>.

MOHAN, J. S. R.; LI, C.-S. Adapting multimedia internet content for universal access. **IEEE Trans. Multimedia**, I, p. 104–114, Março 1999.

MOVIELABS. **MovieLabs specification for enhanced content protection, v1.1**. 2015. Disponível em: <http://movielabs.com/ngvideo/>.

OTHMAN, S. B.; TRAD, A.; YOUSSEF, H. Performance evaluation of encryption algorithm for wireless sensor networks. **2012 International Conference on Information Technology and e-Services**, Março 2012.

PAREIT, D. et al. The history of wimax: A complete survey of the evolution in certification and standardization for ieee 802.16 and wimax. **IEEE Communications Surveys & Tutorials**, v. 14, p. 1183 – 1211, 2012.

PELTONIEMI, J. **Video on Demand Overview**. Finlândia, Janeiro 1995.

PIGATTO, D. F. **Segurança em sistemas embarcados críticos - utilização de criptografia para comunicação segura**. 88 p. Dissertação (Mestrado) — Universidade de São Paulo, 2012.

PIZZOTTI, R. **Video Progressive Scan**. 2019. Disponível em: <https://www.tevepro.com/sistema-entrelacado>.

PURI, A.; CHEN, X.; LUTHRA, A. Video coding using the h.264/mpeg-4 avc compression standard. **Signal Processing: Image Communication**, v. 9, p. 793–849, Outubro 2004.

RAMIREZ, D. **IPTV Security: Protecting High-Value Digital Contents**. 1st. ed. UK: John Wiley and Sons Ltd., 2008. 234 p.

SAMSUNG. **Tizen Studio**. 2018. Disponível em: <http://developer.samsung.com/tv/develop/tools/tizenstudio>.

SHAH, J.; SAXENA, D. V. Video encryption: A survey. **IJCSI International Journal of Computer Science Issues**, v. 8, p. 525–534, 2011.

STALLINGS, W. **Criptografia e segurança de redes: princípios e práticas**. 6. ed. [S.l.]: Pearson Brasil, 2015. 264 p.

TANENBAUM, A. S. **Redes de Computadores**. Rio de Janeiro: Editora Campus, 2003. 945 p.

TCG. **Trusted Platform Module Library. Part 1: Architecture. Family 2.0. Revision 01.16**. Beaverton, Oregon, EUA, Outubro 2014.

VISION, B. **Video Concepts & Terminology**. 2015. Disponível em: <http://www.bubblevision.com/underwater-video/concepts.htm>.

W3C. **Web Crypto API**. 2017. Disponível em: <https://www.w3.org/TR/WebCryptoAPI/>.

WATKINSON, J. **The art of digital video**. 4th. ed. Oxford-UK: Elsevier, 2008. 672 p.

WIEN, M. **High Efficiency Video Coding: Coding Tools and Specification (Signals and Communication Technology)**. 2015th. ed. Berlim: Springer-Verlag Berlin Heidelberg, 2015. 314 p.

WU, Y.; BAO, F. Enriched trusted platform and its application on drm. **Third Asia-Pacific Trusted Infrastructure Technologies Conference**, p. 91–97, 2008.

XIPH.ORG. **Xiph.org Video Test Media [derf's collection]**. 2016. Disponível em: <https://media.xiph.org/video/derf/>.

YANG, Y. et al. Downloadable trusted applications on tizen^{TM} tv. **IEEE International Conference on Consumer Electronics (ICCE)**, 2018.

YU, A.; FENG, D.; LIU, R. Tbdrm: A tpm-based secure drm architecture. **International Conference on Computational Science and Engine**, p. 376–377, 2009.

APÊNDICE A – PRODUÇÃO ACADÊMICA

COSTA, Ricardo de S.; PIGATTO, Daniel F.; FONSECA, Keiko V. O.; ROSA, Marcelo de O.. **Securing Video on Demand Content with SGX: A Decryption Performance Evaluation in Client-Side**. In: SIMPÓSIO BRASILEIRO EM SEGURANÇA DA INFORMAÇÃO E DE SISTEMAS COMPUTACIONAIS (SBSEG), 2018, Natal. Anais do XVIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais. Porto Alegre: Sociedade Brasileira de Computação, oct. 2018 . p. 127 - 140.